Anotações

Projeto em:

NodeJs

React

Baixar:  
NodeJs

Passos:

No terminal usar o comando “npm init –y”

Cria o arquivo package.json

No terminal usar o comando “npm i typescript –D”

No terminal usar o comando “npx tsc –init”

Cria o arquivo tsconfig.json

“procurar no google por tsconfig bases” , buscar a versão do node no git do projeto, copia a configuração do tsconfig e cola no nosso

No terminal usar o comando “npm i @types/node tsx -D”

Integração do typescrip com node

Tsx permite executar o projeto sem antes converter o código para javascript

No terminal usar o comando “npm i fastify”

Framework como o express

Instalar Biomejs (downlod, extensão) verificar no site a instalação certinha ou usar o comando a baixo

Serve para organizar as formatações

No terminal usar o comando “npm i –D –save-exact @biomejs/biome”

Criar um arquivo chamado biome.json

Buscar as configurações melhores pra por, ou usar a do diego3g

Depois usar CTRL+SHIFT+P ou buscar por “>Preferences: Open Workspace Settings (JSON)”

É para configurar alguns ajustes do vscode apenas para esse projeto

Em .vscode/settings.json colocar “{“editor.formatOnSave”: true, “editor.defaultFormatter”: “biomejs.biome”}”

Instalar o docker desktop

Criar um arquivo “docker-compose.yml”

No terminal usar o comando “npm i drizzle-orm”

No terminal usar o comando “npm i drizzle-kit -D”

Criar um arquivo “drizzle.config.ts”

Colocar as configurações do informadas no sita de drizzle em drizzle-kit

Criar arquivo para variáveis de ambiente “.env”

Atualizar o package.json para carregar com as flags de variáveis de ambiente

Instalar a biblioteca “npm i zod”

Criar o arquivo no scr “env.ts”

Instalar o postgres “npm i postgres”

Criar o arquivo de schema em um diretório src/db/schema.ts

Usando o drizzle-orm para criar tabela

Após criar o arquivo para criar tabela

Executar o comando “npx drizzle-kit generate”

Executar o comando “npx drizzle-kit migrate”

Para visualizar usar o comando “npx drizzle-kit studio”

Instalar com comando “npm i @paralleldrive/cuid2”

Para alto incremente das pk

Instalar uma lib de datas “npm i dayjs”

Instalar a lib “npm i fastify-type-provider-zod”

Fazer a configuração igual ao site

Instalações:

npm i fastify-type-provider-zod (

npm i dayjs (para formatação de data)

npm i @paralleldrive/cuid2 (para gerar ids)

npm i postgres (para usar postgres)

npm i zod (

npm i drizzle-kit –D (

npm i drizzle-orm (

npm i -D --save-exact @biomejs/biome (

npm i fastify (

npm i @types/node tsx –D (

npm i typescript -D (

npm init –y (

---

npm i @fastify/cors (

npm create vite@latest (no diretório anterior)(

name

react

typescript

npm i @tanstack/react-query

npm i react-hook-form @hookform/resolvers

Extensões:

Biome

Comment tagged templates (Matt Bierner)

Drizzle ORM snippets

Material Icon

Console ninja

Duvidas:

Interface

Async

Arquitetura de projetos

Front-end

npm create vite@latest (para instalar tudo do vite e depois excluir os desnecessários)

npm i (para instalar as dependências)

Npm i –D @biomejs/biome (colocar as configurações iguais do back-end apenas adicionando em “linter”   
"style": {

        "noNonNullAssertion": "off"

      }

npm install -D tailwindcss postcss autoprefixer

instalar a extenção do tailwind

npm i lucide-react

npm i tailwind-merge

npm i tailwind-variants

npm i @radix-ui/react-radio-group

npm i @radix-ui/react-progress

npm i @radix-ui/react-dialog

ferramenta “transform.tools”

Aqui está uma explicação detalhada sobre cada comando e suas respectivas bibliotecas:

**1. npm init -y**

* **Descrição**: Inicializa um novo projeto Node.js e cria um arquivo package.json com opções padrão.
* **Biblioteca**: **npm** (Node Package Manager), usada para gerenciar dependências em projetos JavaScript.

**2. npm i typescript -D**

* **Descrição**: Instala o TypeScript como uma dependência de desenvolvimento.
* **Biblioteca**: **TypeScript**, uma linguagem de programação que adiciona tipagem estática ao JavaScript.

**3. npx tsc --init**

* **Descrição**: Inicializa o TypeScript no projeto e gera o arquivo tsconfig.json com as configurações básicas.
* **Biblioteca**: **npx** executa pacotes diretamente da linha de comando.

**4. npm i @types/node tsx -D**

* **Descrição**:
  + Instala os tipos do Node.js para o TypeScript (@types/node).
  + Instala o tsx, um executor de arquivos TypeScript, útil para rodar scripts TS sem compilar manualmente.

**5. npm i fastify**

* **Descrição**: Instala o Fastify, um framework web rápido para Node.js.
* **Biblioteca**: **Fastify**, usado para construir APIs e servidores web de forma performática.

**6. npm i -D --save-exact @biomejs/biome**

* **Descrição**: Instala o Biome como uma dependência de desenvolvimento, uma ferramenta para lint, formatação e análise de código.
* **Biblioteca**: **Biome**, similar a ESLint e Prettier, para melhorar a qualidade do código.

**7. npm i drizzle-orm**

* **Descrição**: Instala o Drizzle ORM, uma biblioteca para mapeamento objeto-relacional (ORM).
* **Biblioteca**: **Drizzle ORM**, usado para interagir com bancos de dados de forma eficiente.

**8. npm i drizzle-kit -D**

* **Descrição**: Instala o Drizzle Kit, uma ferramenta de migrações de banco de dados para o Drizzle ORM.
* **Biblioteca**: **Drizzle Kit**, usado para gerenciar migrações de banco de dados em projetos Drizzle ORM.

**9. drizzle.config.ts**

* **Descrição**: Arquivo de configuração onde são definidas as opções de migração e conexão ao banco de dados para o Drizzle ORM e Drizzle Kit.

**10. npm i zod**

* **Descrição**: Instala o Zod, uma biblioteca para validação de dados com suporte a TypeScript.
* **Biblioteca**: **Zod**, usado para validar entradas e garantir tipos corretos de dados.

**11. npm i postgres**

* **Descrição**: Instala o driver Postgres, usado para conectar e interagir com bancos de dados PostgreSQL.
* **Biblioteca**: **Postgres**, um driver para Postgres em Node.js.

**12. npx drizzle-kit generate**

* **Descrição**: Gera as migrações automaticamente com base no modelo de dados definido no Drizzle ORM.

**13. npx drizzle-kit migrate**

* **Descrição**: Executa as migrações de banco de dados, aplicando alterações definidas pelo Drizzle ORM.

**14. npx drizzle-kit studio**

* **Descrição**: Abre uma interface gráfica (studio) para gerenciar o banco de dados e visualizar as migrações, similar ao Prisma Studio.

**15. npm i @paralleldrive/cuid2**

* **Descrição**: Instala o CUID2, uma biblioteca para gerar identificadores únicos (UUIDs) de maneira mais simples e segura.
* **Biblioteca**: **CUID2**, usado para gerar IDs únicos.

**16. npm i dayjs**

* **Descrição**: Instala o Day.js, uma biblioteca para manipulação de datas de forma simples e leve.
* **Biblioteca**: **Day.js**, similar ao Moment.js, mas com foco em ser menor e mais eficiente.

**17. npm i fastify-type-provider-zod**

* **Descrição**: Instala um provedor de tipos que integra o Zod ao Fastify, permitindo validação de esquemas no servidor.
* **Biblioteca**: Integra **Fastify** com **Zod**.

**18. npm i @fastify/cors**

* **Descrição**: Instala o plugin de CORS para o Fastify, permitindo configurar regras de compartilhamento de recursos entre diferentes origens.
* **Biblioteca**: **CORS** (Cross-Origin Resource Sharing), usado para permitir ou restringir requisições de outros domínios.

**19. npm create vite@latest**

* **Descrição**: Cria um novo projeto frontend com **Vite**, uma ferramenta de build rápida para projetos JavaScript modernos, como React ou Vue.js.

**20. npm i @tanstack/react-query**

* **Descrição**: Instala o React Query, uma biblioteca para gerenciamento de estado assíncrono (requisições, cache) em aplicações React.
* **Biblioteca**: **React Query**, usado para lidar com dados que vêm de APIs.

**21. npm i react-hook-form @hookform/resolvers**

* **Descrição**:
  + Instala o React Hook Form, uma biblioteca leve para manipulação de formulários em React.
  + Instala o resolvers, que integra bibliotecas de validação como Zod e Yup ao React Hook Form.

**22. npm install -D tailwindcss postcss autoprefixer**

* **Descrição**:
  + Instala o Tailwind CSS, um framework CSS utilitário para estilização rápida.
  + Instala o PostCSS e o Autoprefixer, que processam o CSS para garantir compatibilidade entre navegadores.

**23. npm i lucide-react**

* **Descrição**: Instala o Lucide React, um pacote de ícones SVG personalizáveis para uso em aplicações React.
* **Biblioteca**: **Lucide**, uma biblioteca de ícones.

**24. npm i tailwind-merge**

* **Descrição**: Instala o Tailwind Merge, uma utilidade que simplifica a combinação de classes do Tailwind CSS para evitar conflitos.
* **Biblioteca**: **Tailwind Merge**, usado para combinar e resolver conflitos de classes CSS.

**25. npm i tailwind-variants**

* **Descrição**: Instala o Tailwind Variants, uma biblioteca para gerenciar variantes condicionais de classes no Tailwind CSS.
* **Biblioteca**: **Tailwind Variants**, facilita a criação de estilos dinâmicos.

**26. npm i @radix-ui/react-radio-group**

* **Descrição**: Instala o componente de grupo de botões de rádio da biblioteca Radix UI para React.
* **Biblioteca**: **Radix UI**, usada para criar componentes acessíveis e personalizáveis.

**27. npm i @radix-ui/react-progress**

* **Descrição**: Instala o componente de barra de progresso da biblioteca Radix UI para React.

**28. npm i @radix-ui/react-dialog**

* **Descrição**: Instala o componente de diálogo da Radix UI, usado para criar janelas modais acessíveis em React.

Esses comandos e bibliotecas são usados para construir aplicações web completas, desde o backend com Fastify, banco de dados com Drizzle ORM e Postgres, até o frontend com React, Vite e Tailwind CSS.