

Integração do FrontEnd com o BackEnd usando o Axios

Integrar o FrontEnd com o BackEnd usando o Axios envolve fazer solicitações HTTP a partir do FrontEnd para endpoints definidos no BackEnd.

Um exemplo de como usar o Axios para fazer uma solicitação de um frontend usando JavaScript para um backend usando Node.js.

- Usando npm

npm install axios

- Usando yarn

yarn add axios

Backend com NestJS e Prisma:

- No backend, usando o NestJS, temos controllers, services, e módulos que gerenciam a lógica de negócios.
- Prisma é uma camada de acesso a dados que pode ser usada para interagir com o banco de dados. Aonde configuramos e utilizamos o Prisma nos services ou controllers para realizar operações de leitura e escrita no banco de dados.

Configuração do Backend NestJS com Prisma

Instale as Dependências:

```
npm install --save @nestjs/core @nestjs/common @nestjs/platform-express @nestjs/cli  
prisma @prisma/client reflect-metadata
```

Configuração do Prisma:

- Execute **npm prisma init** para inicializar o Prisma.
- Configure o arquivo **schema.prisma** conforme necessário.
- Execute **npm prisma migrate dev** para aplicar as migrações.

Configuração do NestJS:

Crie um módulo, um controlador e um serviço

Exemplo de app.controller.ts:

```
import { Controller, Get } from '@nestjs/common';  
import { AppService } from './app.service';  
@Controller()  
export class AppController {  
  constructor(private readonly appService: AppService) {}  
  @Get()  
  getData() {  
    return this.appService.getData();  
  }  
}
```

Exemplo de app.service.ts:

```
import { Injectable } from '@nestjs/common';  
@Injectable()  
export class AppService {  
  getData(): string {  
    return 'Hello from NestJS!';  
  }  
}
```

```
}
```

Configuração do TypeScript:

Certifique-se de ter um arquivo tsconfig.json configurado corretamente para o NestJS.

Inicie o Servidor:

Execute **npm run start:dev** para iniciar o servidor NestJS.

Frontend com Next.js:

- No frontend, usando Next.js, temos páginas, componentes e serviços que gerenciam a interface do usuário.
- Use o Axios para fazer solicitações HTTP para o backend quando necessário. Por exemplo, ao buscar dados para renderizar em uma página Next.js.

Configuração do Frontend (Next.js)

Crie um Projeto Next.js:

```
npx create-next-app my-next-app
```

```
cd my-next-app
```

Instale as Dependências:

```
npm install axios
```

Crie uma Página para Exibir os Dados:

Exemplo de pages/index.tsx:

```
import { useEffect, useState } from 'react';
import axios from 'axios';

export default function Home() {
  const [data, setData] = useState<string>('');

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get('/api/data'); // Substitua isso pelo seu
endpoint real
        setData(response.data);
      } catch (error) {
        console.error('Erro na solicitação para o backend:', error);
      }
    };

    fetchData();
  }, []);

  return (
    <div>
      <h1>Next.js Frontend</h1>
      <p>Data do Backend: {data}</p>
    </div>
  );
}
```

Configuração do TypeScript:

Certifique-se de ter um arquivo tsconfig.json configurado corretamente para o Next.js.

Inicie o Aplicativo Next.js:

Execute **npm run dev** para iniciar o servidor Next.js.