

Guia Rápido: Restaurar Banco e Rodar o Projeto

Este guia mostra como criar o banco de dados PostgreSQL a partir de um dump e rodar um projeto Node.js rapidamente.

1 Criar o banco de dados

Primeiro, abra o terminal ou prompt de comando e crie um banco vazio:

```
createdb -U postgres nomedobanco
```

Substitua `nomedobanco` pelo nome que você deseja dar ao banco.

`postgres` é o usuário padrão do PostgreSQL; se o seu projeto tiver outro usuário, use ele.

2 Restaurar o dump

Em seguida, restaure o arquivo de dump fornecido no banco recém-criado:

```
PGPASSWORD='sua_senha' pg_restore -U postgres -d nomedobanco -v  
/caminho/para/backup_file.dump
```

- `sua_senha` → senha do usuário PostgreSQL.
- `/caminho/para/backup_file.dump` → caminho completo do arquivo de dump no computador.
- `-v` → modo verbose, para mostrar o progresso da restauração.

Ao final deste passo, todas as tabelas e dados estarão disponíveis no banco.

3 Instalar dependências do Node.js

Certifique-se de que os arquivos `package.json` e `package-lock.json` estão no projeto. Então, instale as dependências:

```
npm install
```

Isso vai baixar todos os módulos necessários para o projeto funcionar corretamente.

4 Rodar o projeto

Após instalar as dependências, inicie o servidor:

```
npm run dev
```

O script `dev` deve estar configurado no `package.json` assim:

```
"scripts": {  
  "dev": "nodemon server.js"  
}
```

O servidor Node.js será iniciado, e você poderá acessar o sistema no navegador, geralmente em `http://localhost:3000`. O `nodemon` reinicia o servidor automaticamente sempre que há alterações no código.

5 Resumo rápido das etapas

1. Criar banco → `createdb -U postgres nomedobanco`
2. Restaurar dump → `pg_restore` com o arquivo `.dump`
3. Instalar dependências → `npm install`
4. Rodar projeto → `npm run dev`