

Webhoster - Plataforma de Gestão de Domínios Web

Este projeto consiste em duas partes: uma API construída com LoopBack 4 para gerenciar dados relacionados a planos, clientes, domínios e pagamentos; e uma interface de usuário construída com React Admin para interagir com a API.

Tecnologias Utilizadas

- React: Biblioteca JavaScript ou TypeScript para a construção da interface do usuário.
- LoopBack 4: Framework de backend para a construção da API REST.
- Docker: Plataforma de contêineres para facilitar a implantação e execução do projeto.
- MySQL: Sistema de gerenciamento de banco de dados relacional.

Instruções de Uso

Pré-requisitos

- [Docker](#) instalado e configurado.
- [Git](#)

Passos para Iniciar o Projeto

1. Clone o repositório:

```
git clone https://github.com/inf23dw1g21/inf23dw1g21m3.git
```

2. Navegue até a pasta raiz do projeto:

```
cd webhoster
```

3. Execute o seguinte comando para iniciar o projeto utilizando o Docker Compose:

```
docker-compose up
```

Isso iniciará a API LoopBack 4, o aplicativo React Admin e o banco de dados MySQL.

4. Acesse a interface do React Admin no navegador:

<http://localhost:3006>

A API está disponível em:

<http://localhost:3000>

5. Explore e teste a plataforma de gerenciamento de domínios.

Dados de Acesso (Para Fins de Teste)

LoopBack 4 API

- URL: <http://localhost:3000>
- Usuário: (Não é necessário para testes locais)

React Admin

- URL: <http://localhost:3006>
- Usuário: (Não é necessário para testes locais)
- Senha: (Não é necessário para testes locais)

Observações

- Certifique-se de que as portas 3000, 3006 e 3306 estejam disponíveis no seu ambiente.

Base de Dados

```
`Plano` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `tipo_de_plano` varchar(512) NOT NULL,  
  `periodicidade` varchar(512) NOT NULL,  
  `preco` int NOT NULL,  
  `armazenamento` varchar(512) NOT NULL,  
  `numero_de_contas_email` int NOT NULL,  
  `numero_de_dominios` int NOT NULL,  
  `largura_de_banda` varchar(512) NOT NULL,  
  `fidelizacao` varchar(512) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
`Cliente` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `nome` varchar(512) NOT NULL,  
  `tipo_de_conta` varchar(512) NOT NULL,  
  `numero_fiscal` varchar(512) NOT NULL,  
  `email` varchar(512) NOT NULL,  
  `contacto` varchar(512) NOT NULL,  
  `periodicidade_de_pagamento` varchar(512) NOT NULL,  
  `data_ultimo_pagamento` datetime NOT NULL,  
  `planoId` int DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 ;  
  
`Dominio` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `nome` varchar(512) NOT NULL,
```

```
    `codigo_TLD` varchar(512) NOT NULL,  
    `estado` tinyint(1) NOT NULL,  
    `data_de_inicio` datetime NOT NULL,  
    `data_de_fim` datetime NOT NULL,  
    `clienteId` int DEFAULT NULL,  
    PRIMARY KEY (`id`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 ;  
  
  `Pagamento` (  
    `id` int NOT NULL AUTO_INCREMENT,  
    `timestamp` datetime NOT NULL,  
    `valor` int NOT NULL,  
    `metodo_de_pagamento` varchar(512) NOT NULL,  
    `numero_de_transacao` varchar(512) NOT NULL,  
    `clienteId` int DEFAULT NULL,  
    PRIMARY KEY (`id`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 ;
```

Nota: Este projeto é destinado apenas para fins educacionais e de teste. Não é recomendado para ambientes de produção sem configurações adicionais de segurança.