

Digital Tasks

Projeto Backend: CRUD de Usuários e Tarefas com Permissões

Visão Geral

Este projeto envolve a criação de um backend para gerenciar usuários e tarefas com operações CRUD e um sistema de permissões utilizando JWT para autenticação e autorização. O uso de banco de dados e a criação de um frontend são opcionais.

Requisitos do Projeto

1. CRUD de Usuários:

- **Criar:** Endpoint para adicionar novos usuários.
- **Ler:** Endpoint para listar e obter detalhes de usuários.
- **Atualizar:** Endpoint para atualizar informações de usuários.
- **Deletar:** Endpoint para remover usuários.

2. CRUD de Tarefas:

- **Criar:** Endpoint para adicionar novas tarefas.
- **Ler:** Endpoint para listar e obter detalhes de tarefas.
- **Atualizar:** Endpoint para atualizar informações de tarefas.
- **Deletar:** Endpoint para remover tarefas.

3. Sistema de Permissões:

- Implementar autenticação e autorização usando JWT.
- Usuários devem ter diferentes papéis (e.g., ADMIN, USER).
- Restringir acesso a determinadas ações com base no papel do usuário.

4. Banco de Dados (Opcional):

5. Frontend (Opcional):

Definições de Pronto

Para que o projeto seja considerado completo, os seguintes critérios devem ser atendidos:

1. **Repositório no GitHub:**

- Projeto hospedado em um repositório no GitHub.
- Todos os membros da equipe adicionados como colaboradores.

2. **Branches e Merges:**

- Pelo menos 15 branches diferentes mergeadas na branch principal.
- Cada branch deve representar uma funcionalidade ou correção distinta.

3. **Documentação no README:**

- `README.md` bem documentado contendo:
 - **Descrição do Projeto**
 - **Instruções de Instalação**
 - **Instruções de Uso**
 - **Tecnologias Utilizadas**
 - **Contribuição**
 - **Autores**

4. **Estrutura do Projeto:**

```
projeto/  
|  
├─ controllers/  
|   ├─ AuthController.js  
|   ├─ TaskController.js  
|   └─ UserController  
|  
|  
├─ models/  
|   ├─ User.js  
|   └─ Task.js  
|  
├─ middleware/  
|   ├─ authMiddleware.js  
|   └─ roleMiddleware.js
```

```
├── errorHandlerMiddleware.js
├── repositories/
│   ├── UserRepository.js
│   └── TaskRepository.js
├── routes/
│   ├── authRoutes.js
│   ├── userRoutes.js
│   └── taskRoutes.js
├── services/
│   ├── AuthService.js
│   ├── UserService.js
│   └── TaskService.js
├── app.js
└── package.json
```

Regras de Negócio

Usuários

1. Criação de Usuário:

- Um novo usuário deve ter os seguintes campos obrigatórios:
 - `nome` : Nome do usuário.
 - `email` : Email do usuário (único).
 - `senha` : Senha do usuário.
 - `papel` : Papel do usuário (ADMIN, USER).

2. Atualização de Usuário:

- Um usuário pode atualizar suas próprias informações, exceto seu papel.
- Somente usuários com papel `ADMIN` podem atualizar os papéis de outros usuários.

3. Remoção de Usuário:

- Um usuário pode remover sua própria conta.
- Somente usuários com papel `ADMIN` podem remover outras contas.

4. Listagem de Usuários:

- Somente usuários com papel `ADMIN` podem listar todos os usuários.

Autenticação e Autorização

1. Login:

- Os usuários devem se autenticar usando email e senha.
- Após autenticação bem-sucedida, um token JWT será gerado e retornado ao usuário.

2. Proteção de Rotas:

- Todas as rotas, exceto a rota de login, devem ser protegidas por autenticação JWT.
- Usuários devem incluir o token JWT no header das requisições para acessar as rotas protegidas.

Tarefas

1. Criação de Tarefas:

- Qualquer usuário autenticado pode criar uma nova tarefa.
- Uma tarefa deve ter os seguintes campos obrigatórios:
 - `titulo` : Título da tarefa.
 - `descricao` : Descrição da tarefa.
 - `status` : Status da tarefa (e.g., PENDENTE, CONCLUÍDA).

2. Atualização de Tarefas:

- Um usuário pode atualizar as tarefas que ele próprio criou.
- Usuários com papel `ADMIN` podem atualizar qualquer tarefa.

3. Remoção de Tarefas:

- Um usuário pode remover as tarefas que ele próprio criou.
- Usuários com papel `ADMIN` podem remover qualquer tarefa.

4. Listagem de Tarefas:

- Qualquer usuário autenticado pode listar todas as tarefas.

Validação de Dados

1. Usuários:

- O email deve ser único e válido.
- A senha deve ter no mínimo 6 caracteres.

2. Tarefas:

- O título deve ter no mínimo 3 caracteres e no máximo 100 caracteres.
- A descrição deve ter no mínimo 10 caracteres.

Regras de Permissão

1. ADMIN:

- Pode criar, ler, atualizar e deletar qualquer usuário.
- Pode criar, ler, atualizar e deletar qualquer tarefa.

2. USER:

- Pode criar, ler e atualizar suas próprias informações, exceto o papel.
- Pode deletar sua própria conta.
- Pode criar, ler, atualizar e deletar suas próprias tarefas.
- Não pode listar todos os usuários.

Exemplos de Fluxo de Usuário

1. Fluxo de Criação de Usuário:

- O usuário envia uma requisição para criar uma conta com seu nome, email e senha.
- O sistema verifica se o email é único e cria o usuário com papel padrão `USER`.

2. Fluxo de Login:

- O usuário envia uma requisição de login com email e senha.
- O sistema autentica o usuário e retorna um token JWT válido por 1 hora.

3. Fluxo de Criação de Tarefa:

- O usuário autenticado envia uma requisição para criar uma tarefa com título e descrição.
- O sistema cria a tarefa associada ao usuário autenticado.

4. **Fluxo de Atualização de Tarefa:**

- O usuário autenticado envia uma requisição para atualizar uma tarefa que ele criou.
- O sistema verifica se o usuário é o criador da tarefa e realiza a atualização.

5. **Fluxo de Remoção de Tarefa:**

- O usuário autenticado envia uma requisição para remover uma tarefa que ele criou.
- O sistema verifica se o usuário é o criador da tarefa e realiza a remoção.

Boa sorte e bom trabalho a todos!