

Documentação da API

Instalação e Configuração

1. Pré-requisitos:

- Node.js 12.x ou superior
- MongoDB instalado ou uma conexão para MongoDB Atlas

2. Instalação:

- Clone o repositório do projeto.
- No terminal, navegue até a raiz do projeto.
- Execute `npm install` para instalar as dependências necessárias listadas em `package.json`.

3. Executando o projeto localmente:

- Para iniciar o servidor, use `node index.js` ou `npm start` se configurado.
- O servidor estará disponível em `http://localhost:3000`.

Estrutura de Arquivos

- `public/` - Contém arquivos HTML para cadastro, login, listagem e criação de projetos.
- `index.js` - Arquivo principal do servidor.
- `.gitignore` - Arquivos e pastas ignorados pelo controle de versão Git.
- `package.json` e `package-lock.json` - Informações de dependências do Node.js.

Rotas da API

Rotas Públicas

`GET /cadastro`

- Retorna a página de cadastro (`cadastro.html`).

`GET /login`

- Retorna a página de login (`login.html`).

Rotas de Autenticação

POST `/auth/register`

- **Descrição:** Registra um novo usuário.
- **Parâmetros de entrada:** `nome` (string), `email` (string), `senha` (string).
- **Exemplo de Entrada:**

```
json
Copiar código
{
  "nome": "Usuário Teste",
  "email": "teste@exemplo.com",
  "senha": "senha123"
}
```

- **Resposta de Sucesso:** Redireciona para `/login` .

POST `/auth/login`

- **Descrição:** Realiza o login de um usuário.
- **Parâmetros de entrada:** `email` (string), `senha` (string).
- **Exemplo de Entrada:**

```
json
Copiar código
{
  "email": "teste@exemplo.com",
  "senha": "senha123"
}
```

- **Resposta de Sucesso:** Retorna um token JWT.

```
json
Copiar código
{
```

```
{
  "token": "eyJhbGciOiJIUzI1..."
}
```

Rotas de Projetos

GET /listagem-projetos

- Retorna a página de listagem de projetos (`listagem-projetos.html`).

GET /novo-projeto

- Retorna a página para criar um novo projeto (`novo-projeto.html`).

POST /projeto

- **Descrição:** Cria um novo projeto.
- **Autenticação:** Token JWT necessário.
- **Parâmetros de entrada:** `nome` (string), `descricao` (string).
- **Exemplo de Entrada:**

```
json
Copiar código
{
  "nome": "Novo Projeto",
  "descricao": "Descrição do projeto"
}
```

- **Resposta de Sucesso:**

```
json
Copiar código
{
  "status": 200,
  "message": "Projeto criado com sucesso!"
}
```

PUT /projeto/:id

- **Descrição:** Edita um projeto existente pelo `id` .

- **Autenticação:** Token JWT necessário.
- **Parâmetros de entrada:** `nome` (string), `descricao` (string).
- **Exemplo de Entrada:**

```
json
Copiar código
{
  "nome": "Projeto Atualizado",
  "descricao": "Nova descrição"
}
```

- **Resposta de Sucesso:**

```
json
Copiar código
{
  "status": 200,
  "message": "Projeto editado com sucesso!"
}
```

GET /projetos

- **Descrição:** Retorna todos os projetos.
- **Autenticação:** Token JWT necessário.
- **Resposta:** Lista de projetos em formato JSON.

GET /projeto/:id

- **Descrição:** Retorna um projeto específico pelo `id`.
- **Autenticação:** Token JWT necessário.

DELETE /projetos/:id

- **Descrição:** Exclui um projeto pelo `id`.
- **Autenticação:** Token JWT necessário. Apenas administradores podem excluir.

Rotas Temporárias (Dados em Memória)

POST /temp/:collection

- Adiciona um item a uma coleção de dados em memória (ex: `logs`, `progresso`).
- Autenticação necessária.

GET /temp/:collection

- Retorna os itens de uma coleção de dados em memória.
- Autenticação necessária.

DELETE /temp/:collection/:id

- Remove um item de uma coleção de dados em memória pelo `id`.
- Autenticação necessária.

Exemplo de Configuração do MongoDB

Substitua a string de conexão em `mongoose.connect` com as credenciais adequadas para seu banco MongoDB.

Observações

- O token JWT é passado nos headers das requisições como `Authorization`.
- O middleware `autenticarToken` protege rotas que requerem autenticação. O `verificarAdmin` restringe operações para usuários com papel de `admin`.