

Swagger

Descrição Técnica do Back-End

O backend do projeto é estruturado de forma modular, com separação clara de responsabilidades entre rotas, controllers, middlewares e utilitários. A aplicação é desenvolvida utilizando **Node.js** com **Express** e toda a API está documentada de forma interativa utilizando o **Swagger UI**, acessível pelo endereço:

“ [http://localhost:\\${PORT}/api/v1/swagger](http://localhost:${PORT}/api/v1/swagger) onde **PORT** é definida pela variável de ambiente `API_PORT` ou assume o valor padrão **3006**.

Lembre-se de ter o projeto rodando no docker para funcionar!

Estrutura de Diretórios:

```
/api/  
├─ src/  
│ └─ controller/      # [] Controladores responsáveis pela lógica de cada domínio.  
│   └─ otp/          # Envio de OTPs e recuperação de senha  
│   └─ patrimonio/  
│   └─ setor/  
│   └─ usuario/  
│  
│ └─ docs/           # [] Scripts de geração e de documentação Swagger.  
│   └─ compare-docs.ts  
│   └─ generate-docs.ts  
│   └─ swagger.ts  
│   └─ swaggerSchemas.ts  
│
```

```
| └─ middlewares/      # 🌀 Middlewares diversos, incluindo o multer para uploads
| | └─ authFacade/
| | └─ basicAuth/
| | └─ checkSetor/
| | └─ multer/
|
| └─ router/          # 📁 Definição de todas as rotas do sistema organizadas por domínio
| | └─ otp/
| | └─ patrimonio/
| | └─ setor/
| | └─ usuario/
| | └─ router.ts      # Roteador principal que importa e agrupa os módulos acima
|
| └─ utils/           # 📁 Funções auxiliares e utilitários reutilizáveis.
| | └─ administrador/
| | └─ email/
| | └─ otp/           # gerar OTP
| | └─ qrcode/
| | └─ types/
| | └─ validations/   # Validação cpf
| | └─ expirarSetor.ts # Validação de expiração do setor
|
| └─ app.ts           # 📁 Ponto de entrada da aplicação e inicialização dos serviços
```

Documentação Interativa com Swagger

A API conta com documentação completa utilizando **Swagger**, onde é possível:

- Visualizar **todos os endpoints disponíveis**
- Ver os **parâmetros esperados** por cada rota
- Testar requisições diretamente pela interface
- Visualizar exemplos de **respostas de sucesso e erro**

Exemplo de rotas documentadas:

- **POST /api/v1/patrimonio/cadastrar**
permite que um usuário com a devida permissão cadastre um novo patrimônio no sistema.
- **PUT /api/v1/patrimonio/editar**
Permite que um usuário com a devida permissão edite um patrimônio.
- **GET /api/v1/usuario/buscar**
Retorna os usuários cadastrados.
- **POST /api/v1/setor/cadastrar**
Permite ao usuário com a devida permissão cadastrar um novo setor no sistema.

Essas e outras rotas estão organizadas por grupos lógicos (Patrimônio, Usuário, Setor) dentro do Swagger UI.

Upload de Arquivos com Multer

A aplicação utiliza o middleware **Multer** para lidar com uploads de arquivos (como csv), salvando-os em diretórios específicos com nomes padronizados conforme o campo da requisição.

Revision #6

Created 22 October 2025 14:05:10 by GuilhermeBDias

Updated 23 October 2025 12:07:24 by GuilhermeBDias