

# API

## Documentação da API

Sistema backend para gerenciar usuários e projetos de clientes de marketing digital via API REST.

v1.0.14

### User Routes

#### POST /register

- **Descrição:** Registra um novo usuário.

- **Requisição:**

- Corpo da requisição ( **JSON** ):

```
{ "name": "string", "email": "string", "password": "string" }
```

- **Parâmetros:**

- **name** : Nome do usuário.
    - **email** : Endereço de e-mail do usuário.
    - **password** : Senha do usuário (deve ser segura).

- **Resposta:**

- **Status 201:** Usuário criado com sucesso.

- Exemplo de resposta:

```
{ "status": 201, "data": { "token": "jwt-token-aqui" }, "message": "Usuário criado com sucesso." }
```

- **Status 400:** Usuário já existe.

- Exemplo de resposta:

```
{ "status": 400, "message": "Usuário já existe."}
```

- **Status 500:** Erro interno do servidor.

## POST /login

- **Descrição:** Faz o login de um usuário.

- **Requisição:**

- Corpo da requisição ( **JSON** ):

```
{ "email": "string", "password": "string"}
```

- **Parâmetros:**

- **email** : Endereço de e-mail do usuário.
- **password** : Senha do usuário.

- **Resposta:**

- **Status 200:** Login bem-sucedido e token gerado.

- Exemplo de resposta:

```
{ "status": 200, "data": { "token": "jwt-token-aqui" }}
```

- **Status 400:** Senha incorreta.

- Exemplo de resposta:

```
{ "status": 400, "message": "Senha incorreta."}
```

- **Status 404:** Usuário não encontrado.

- Exemplo de resposta:

```
{ "status": 404, "message": "Usuário não encontrado."}
```

- **Status 500:** Erro interno do servidor.

# Client Routes

## GET /

- **Descrição:** Retorna todos os clientes.
- **Autenticação:** Requer autenticação (middleware `authMiddleware`).
- **Resposta:**
  - **Status 200:** Lista de clientes.

- Exemplo de resposta:

```
{ "status": 200, "data": [ { "id": 1, "name": "Cliente 1",  
"email": "cliente1@example.com" }, { "id": 2, "name": "Cli  
ente 2", "email": "cliente2@example.com" } ] }
```

- **Status 400:** Nenhum cliente encontrado.

- Exemplo de resposta:

```
{ "status": 400, "message": "Nenhum cliente encontrado." }
```

- **Status 500:** Erro interno do servidor.

## GET /:id

- **Descrição:** Retorna um cliente específico pelo ID.
- **Autenticação:** Requer autenticação (middleware `authMiddleware`).
- **Parâmetros:**

- `id`: ID do cliente.

- **Resposta:**
  - **Status 200:** Cliente encontrado.

- Exemplo de resposta:

```
{ "status": 200, "data": { "id": 1, "name": "Cliente 1", "email":
```

```
"cliente1@example.com" }}
```

- **Status 400:** Cliente não encontrado.

- Exemplo de resposta:

```
{ "status": 400, "message": "Cliente não encontrado."}
```

- **Status 500:** Erro interno do servidor.

## POST /

- **Descrição:** Cria um novo cliente.
- **Autenticação:** Requer autenticação (middleware `authMiddleware` ).
- **Requisição:**

- Corpo da requisição ( `JSON` ):

```
{ "name": "string", "email": "string"}
```

- **Parâmetros:**

- `name` : Nome do cliente.
- `email` : E-mail do cliente.

- **Resposta:**

- **Status 201:** Cliente criado com sucesso.

- Exemplo de resposta:

```
{ "status": 201, "message": "Cliente criado com sucesso.", "data": { "id": 1, "name": "Cliente 1", "email": "cliente1@example.com" }}
```

- **Status 400:** Cliente já existe.

- Exemplo de resposta:

```
{ "status": 400, "message": "Cliente já existe."}
```

- **Status 500:** Erro interno do servidor.

## PUT /:id

- **Descrição:** Atualiza as informações de um cliente específico.
- **Autenticação:** Requer autenticação (middleware `authMiddleware`).
- **Parâmetros:**

- `id`: ID do cliente.

- **Requisição:**

- Corpo da requisição (`JSON`):

```
{ "name": "string", "email": "string"}
```

- **Parâmetros:**

- `name`: Nome do cliente.
- `email`: E-mail do cliente.

- **Resposta:**

- **Status 201:** Cliente atualizado com sucesso.

- Exemplo de resposta:

```
{ "status": 201, "message": "Cliente atualizado com sucesso.", "data": { "id": 1, "name": "Cliente 1", "email": "cliente1@example.com" }}
```

- **Status 500:** Erro ao atualizar o cliente.

- Exemplo de resposta:

```
{ "status": 500, "message": "Erro ao atualizar o cliente."}
```

## DELETE /:id

- **Descrição:** Exclui um cliente pelo ID.
- **Autenticação:** Requer autenticação (middleware `authMiddleware`).
- **Parâmetros:**
  - `id`: ID do cliente.
- **Resposta:**
  - **Status 201:** Cliente excluído com sucesso.
    - Exemplo de resposta:

```
{ "status": 201, "message": "Cliente excluído com sucesso."}
```
  - **Status 400:** Cliente não encontrado.
    - Exemplo de resposta:

```
{ "status": 400, "message": "Cliente não encontrado."}
```
  - **Status 500:** Erro interno do servidor.

## Project Routes

### GET /

- **Descrição:** Retorna todos os projetos.
- **Autenticação:** Requer autenticação (middleware `authMiddleware`).
- **Resposta:**
  - **Status 200:** Lista de projetos.
    - Exemplo de resposta:

```
{ "status": 200, "data": [ { "id": 1, "name": "Projeto 1",  
  "description": "Descrição do Projeto 1" }, { "id": 2, "name": "Projeto 2",  
  "description": "Descrição do Projeto 2" } ] }
```

- **Status 400:** Nenhum projeto encontrado.

- Exemplo de resposta:

```
{ "status": 400, "message": "Nenhum projeto encontrado."}
```

- **Status 500:** Erro interno do servidor.

## GET /:id

- **Descrição:** Retorna um projeto específico pelo ID.
- **Autenticação:** Requer autenticação (middleware `authMiddleware`).
- **Parâmetros:**
  - `id`: ID do projeto.
- **Resposta:**

- **Status 200:** Projeto encontrado.

- Exemplo de resposta:

```
{ "status": 200, "data": { "id": 1, "name": "Projeto 1", "description": "Descrição do Projeto 1" }}
```

- **Status 404:** Projeto não encontrado.

- Exemplo de resposta:

```
{ "status": 404, "message": "Projeto não encontrado."}
```

- **Status 500:** Erro interno do servidor.

## POST /

- **Descrição:** Cria um novo projeto.
- **Autenticação:** Requer autenticação (middleware `authMiddleware`).
- **Requisição:**
  - Corpo da requisição ( `JSON` ):

```
{ "name": "string", "description": "string"}
```

- **Parâmetros:**
  - `name` : Nome do projeto.
  - `description` : Descrição do projeto.
- **Resposta:**
  - **Status 201:** Projeto criado com sucesso.
    - Exemplo de resposta:

```
{ "status": 201, "message": "Projeto criado com sucesso.", "data": { "id": 1, "name": "Projeto 1", "description": "Descrição do Projeto 1" }}
```

- **Status 500:** Erro ao criar o projeto.
  - Exemplo de resposta:

```
{ "status": 500, "message": "Erro ao criar o projeto."}
```

## **PUT /:id**

- **Descrição:** Atualiza as informações de um projeto específico.
- **Autenticação:** Requer autenticação (middleware `authMiddleware` ).
- **Parâmetros:**
  - `id` : ID do projeto.
- **Requisição:**
  - Corpo da requisição ( `JSON` ):

```
{ "name": "string", "description": "string"}
```

- **Parâmetros:**



- `name` : Nome do projeto.
- `description` : Descrição do projeto.

- **Resposta:**

- **Status 201:** Projeto atualizado com sucesso.

- Exemplo de resposta:

```
{ "status": 201, "message": "Projeto atualizado com sucesso.", "data": { "id": 1, "name": "Projeto 1", "description": "Descrição do Projeto 1" }}
```

- **Status 500:** Erro ao atualizar o projeto.

- Exemplo de resposta:

```
{ "status": 500, "message": "Erro ao atualizar o projeto."}
```

## DELETE /:id

- **Descrição:** Exclui um projeto pelo ID.
- **Autenticação:** Requer autenticação (middleware `authMiddleware`).
- **Parâmetros:**
  - `id` : ID do projeto.

- **Resposta:**

- **Status 201:** Projeto excluído com sucesso.

- Exemplo de resposta:

```
{ "status": 201, "message": "Projeto excluído com sucesso."}
```

- **Status 400:** Projeto não encontrado.

- Exemplo de resposta:

```
{ "status": 400, "message": "Projeto não encontrado."}
```

- **Status 500:** Erro interno do servidor.
- 

## Resumo

As rotas fornecem operações CRUD (criar, ler, atualizar, excluir) para os projetos. Assegure-se de fornecer a autenticação através do middleware `authMiddleware` para todas as operações.