

Documentação das Rotas de Clientes

1. Login de Cliente

Endpoint: /login

Método: POST

Descrição: Autentica um cliente e retorna um token JWT.

Parâmetros de Requisição:

```
{
  "email": "cliente@example.com",
  "senha": "senhaSegura123"
}
```

```
{
  "email": "cliente@example.com",
  "senha": "senhaSegura123"
}
```

Resposta de Sucesso:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "id": 1
}
```

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "id": 1
}
```

Erros Possíveis:

- 400 Bad Request: Email e senha são obrigatórios
- 400 Bad Request: Email ou senha incorretos
- 500 Internal Server Error: Erro ao verificar email

2. Buscar Todos os Clientes

Endpoint: /clientes

Método: GET

Descrição: Retorna uma lista de todos os clientes. Requer autenticação de administrador.

Autenticação: Bearer Token (Admin)

Exemplo de Requisição:

```
GET /clientes
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

```
GET /clientes
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Resposta de Sucesso:

```
[
  {
    "id": 1,
    "nome": "Cliente Exemplo",
    "telefone": "(11) 98765-4321",
    "especialidade": "Cardiologia",
    "email": "cliente@example.com",
    "senha": "hashedPassword",
    "rua_avenida": "Rua Exemplo",
    "numero": "123",
    "complemento": "Apto 101",
    "bairro": "Centro",
    "cidade": "São Paulo",
    "estado": "SP",
    "cep": "01001-000",
    "pais": "Brasil",
    "cpf": "123.456.789-09",
    "cnpj": null,
    "crm": "123456"
  }
]
```

```
[
  {
    "id": 1,
    "nome": "Cliente Exemplo",
    "telefone": "(11) 98765-4321",
    "especialidade": "Cardiologia",
    "email": "cliente@example.com",
    "senha": "hashedPassword",
    "rua_avenida": "Rua Exemplo",
    "numero": "123",
    "complemento": "Apto 101",
    "bairro": "Centro",
    "cidade": "São Paulo",
    "estado": "SP",
    "cep": "01001-000",
    "pais": "Brasil",
    "cpf": "123.456.789-09",
    "cnpj": null,
    "crm": "123456"
  }
]
```

Erros Possíveis:

- 500 Internal Server Error: Erro ao buscar clientes

3. Cadastrar um Novo Cliente

Endpoint: /clientes

Método: POST

Descrição: Cadastra um novo cliente.

Parâmetros de Requisição:

```
{
  "nome": "Novo Cliente",
  "telefone": "(11) 98765-4321",
  "especialidade": "Pediatria",
  "email": "novocliente@example.com",
  "senha": "senhaSegura123",
  "rua_avenida": "Rua Nova",
```

```
"numero": "456",  
"complemento": "Apto 202",  
"bairro": "Bairro Novo",  
"cidade": "São Paulo",  
"estado": "SP",  
"cep": "02002-000",  
"pais": "Brasil",  
"cpf": "987.654.321-00",  
"cnpj": null,  
"crm": "654321"  
}
```

```
{  
  "nome": "Novo Cliente",  
  "telefone": "(11) 98765-4321",  
  "especialidade": "Pediatria",  
  "email": "novocliente@example.com",  
  "senha": "senhaSegura123",  
  "rua_avenida": "Rua Nova",  
  "numero": "456",  
  "complemento": "Apto 202",  
  "bairro": "Bairro Novo",  
  "cidade": "São Paulo",  
  "estado": "SP",  
  "cep": "02002-000",  
  "pais": "Brasil",  
  "cpf": "987.654.321-00",  
  "cnpj": null,  
  "crm": "654321"  
}
```

Resposta de Sucesso:

- 201 Created: Cliente cadastrado com sucesso

Erros Possíveis:

- 400 Bad Request: Campos obrigatórios não preenchidos
 - 400 Bad Request: A senha deve ter pelo menos 6 caracteres
 - 400 Bad Request: CPF inválido
 - 400 Bad Request: CNPJ inválido
 - 400 Bad Request: Campo CRM é obrigatório
 - 500 Internal Server Error: Erro ao inserir cliente
-

4. Buscar um Cliente Específico

Endpoint: `/clientes/:id`

Método: GET

Descrição: Retorna os detalhes de um cliente específico. Requer autenticação de administrador.

Autenticação: Bearer Token (Admin)

Parâmetros de URL:

- `id`: ID do cliente

Exemplo de Requisição:

```
GET /clientes/1
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

```
GET /clientes/1
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Resposta de Sucesso:

```
{
  "id": 1,
  "nome": "Cliente Exemplo",
  "telefone": "(11) 98765-4321",
  "especialidade": "Cardiologia",
  "email": "cliente@example.com",
  "senha": "hashedPassword",
  "rua_avenida": "Rua Exemplo",
  "numero": "123",
  "complemento": "Apto 101",
  "bairro": "Centro",
  "cidade": "São Paulo",
  "estado": "SP",
  "cep": "01001-000",
  "pais": "Brasil",
  "cpf": "123.456.789-09",
  "cnpj": null,
  "crm": "123456"
}
```

```
{
  "id": 1,
  "nome": "Cliente Exemplo",
  "telefone": "(11) 98765-4321",
  "especialidade": "Cardiologia",
  "email": "cliente@example.com",
  "senha": "hashedPassword",
  "rua_avenida": "Rua Exemplo",
  "numero": "123",
  "complemento": "Apto 101",
  "bairro": "Centro",
  "cidade": "São Paulo",
  "estado": "SP",
  "cep": "01001-000",
  "pais": "Brasil",
  "cpf": "123.456.789-09",
  "cnpj": null,
  "crm": "123456"
}
```

Erros Possíveis:

- 404 Not Found: Cliente não encontrado
- 500 Internal Server Error: Erro ao buscar cliente

5. Atualizar um Cliente Existente

Endpoint: /clientes/:id

Método: PUT

Descrição: Atualiza os detalhes de um cliente existente. Requer autenticação.

Autenticação: Bearer Token

Parâmetros de URL:

- id: ID do cliente

Parâmetros de Requisição:

```
{
  "nome": "Cliente Atualizado",
```

```
"telefone": "(11) 98765-4321",
"especialidade": "Neurologia",
"email": "clienteatualizado@example.com",
"senha": "novaSenhaSegura123",
"rua_avenida": "Rua Atualizada",
"numero": "789",
"complemento": "Apto 303",
"bairro": "Bairro Atualizado",
"cidade": "São Paulo",
"estado": "SP",
"cep": "03003-000",
"pais": "Brasil",
"cpf": "123.456.789-09",
"cnpj": null,
"crm": "789123"
}
```

```
{
  "nome": "Cliente Atualizado",
  "telefone": "(11) 98765-4321",
  "especialidade": "Neurologia",
  "email": "clienteatualizado@example.com",
  "senha": "novaSenhaSegura123",
  "rua_avenida": "Rua Atualizada",
  "numero": "789",
  "complemento": "Apto 303",
  "bairro": "Bairro Atualizado",
  "cidade": "São Paulo",
  "estado": "SP",
  "cep": "03003-000",
  "pais": "Brasil",
  "cpf": "123.456.789-09",
  "cnpj": null,
  "crm": "789123"
}
```

Resposta de Sucesso:

- 200 OK: Cliente atualizado com sucesso

Erros Possíveis:

- 400 Bad Request: Campos obrigatórios não preenchidos
- 400 Bad Request: A senha deve ter pelo menos 6 caracteres
- 400 Bad Request: CPF inválido
- 400 Bad Request: CNPJ inválido
- 500 Internal Server Error: Erro ao atualizar cliente

6. Deletar um Cliente

Endpoint: `/clientes/:id`

Método: `DELETE`

Descrição: Deleta um cliente específico. Requer autenticação de administrador.

Autenticação: Bearer Token (Admin)

Parâmetros de URL:

- `id`: ID do cliente

Exemplo de Requisição:

```
DELETE /clientes/1
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

```
DELETE /clientes/1
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Resposta de Sucesso:

- `200 OK`: Cliente deletado com sucesso

Erros Possíveis:

- `500 Internal Server Error`: Erro ao deletar cliente