

Projeto Ciência de dados:

Descrição Geral

A API de Pets oferece funcionalidades para gerenciar uma clínica de PetShop, permitindo a criação, edição, remoção, busca e listagem de Pets. Ela foi projetada para facilitar o gerenciamento de pets e pode ser integrada ao sistema web armazenando os dados em um banco NoSQL(MongoDB).

Formato de Respostas:

Todas as respostas da API são fornecidas no formato JSON:

```
_id: ObjectId('6646836e9e8c1e529eb6fe39')
titulo: "lucas"
veterinario: "lucas"
petSistter: "lucas"
```

Endpoints

1. Criar Pet

- a. **Método:** POST
- b. **Rota:** /pets
- c. **Descrição:** Adiciona um Pet ao sistema do Petshop.
- d. **Parâmetros de Corpo(JSON)**
 - `_id`: (string) - auto generated Mongo.
 - `titulo`: (string) - nome do tipo do pet.
 - `veterinário`: (string) - veterinário responsável pelo pet.
 - `petSitter`: (string) - cuidador do pet.
- e. **Exemplo de requisição:**

```
{
  "titulo": "Papagaio",
  "veterinario": "Virgulino",
  "petSistter": "Jadson"
}
```

f. Exemplo de resposta

```
"message": "Pet cadastrado com sucesso!",
"pets": {
  "titulo": "Papagaio",
  "veterinario": "Virgulino",
  "petSistter": "Jadson",
  "_id": "66f345f0f538d1b96ba59e9d"
```

2. Buscar Pet

a. Método: GET

b. Rota: /pets

c. Descrição: Recupera todos os Pets cadastrados.

d. Parâmetros de Consulta(JSON)

- _id: (string) - auto generated Mongo.
- titulo: (string) - Opcional
- veterinário: (string) - Opcional
- petSitter: (string) - Opcional

e. Exemplo de requisição:

```
http://localhost:3000/pets
```

f. Exemplo de resposta

```
{
  "_id": "6646813c9e8c1e529eb6fe33",
  "titulo": "CAT",
  "veterinario": "carlos",
  "petSistter": "Lucas"
},
{
  "_id": "6646836e9e8c1e529eb6fe39",
  "titulo": "lucas",
  "veterinario": "lucas",
  "petSistter": "lucas"
},
{
  "_id": "6648bd71b91bda0897d80670",
  "titulo": "burro",
  "veterinario": "lucas",
  "petSistter": "igor"
},
{
```

3. Editar Pet

- a. **Método:** PUT
- b. **Rota:** /pets/{id}
- c. **Descrição:** Edita o pet com o {id} específico.
- d. **Parâmetros de Caminho:**
 - `_id`: (string) - Obrigatório
- e. **Parâmetros de Corpo(JSON)**
 - `titulo`: (string) - Opcional
 - `veterinario`: (string) - Opcional
 - `petSistter`: (string) - Opcional
- f. **Exemplo de requisição:**

```
{
  "titulo": "macaco",
  "veterinario": "carlos",
  "petSistter": "Lucas"
}
```

- g. **Exemplo de resposta**

```
{
  "message": "Pet atualizado com sucesso!",
  "pet": {
    "_id": "6646813c9e8c1e529eb6fe33",
    "titulo": "macaco",
    "veterinario": "carlos",
    "petSistter": "Lucas"
  }
}
```

4. Deletar Pet

- a. **Método:** DELETE
- b. **Rota:** /pets/{id}
- c. **Descrição:** Excluir os Pets cadastrados com o {id} específico.
- d. **Parâmetros de Caminho:**
 - `_id`: (string) - Obrigatório.
- e. **Exemplo de requisição:**

```
DELETE  http://localhost:3000/pets/66f345f0f538d1b96ba59e9d
```

- f. **Exemplo de resposta**

```
{
  "message": "Pet excluído com sucesso!"
}
```

Códigos de Resposta

- **200 OK:** Requisição bem-sucedida.
- **201 Created:** Recurso criado com sucesso.
- **400 Bad Request:** A requisição contém erros (ex.: parâmetros faltando ou inválidos).
- **401 Unauthorized:** Autenticação necessária ou inválida.
- **404 Not Found:** Recurso não encontrado.
- **500 Internal Server Error:** Erro no servidor.