

Apis Doc

Documentation de l'API Pets-API-Doc

GET - Ping de l'API

- **Description** : Vérifie la connectivité avec l'API.
- **URL** : `/commands/ping`
- **Exemple d'utilisation** :

```
curl -X GET http://localhost:8084/commands/ping
```

- **Réponse attendue** :

```
{  
  "status": "OK"  
}
```

GET - Liste de tous les animaux

- **Description** : Renvoie la liste de tous les animaux enregistrés.
- **URL** : `/pets`
- **Exemple d'utilisation** :

```
curl -X GET http://localhost:8084/pets
```

- **Réponse attendue** :

```
[
  {
    "id": 1,
    "name": "Rex",
    "age": 5,
    "owner": "Jack"
  },
  ...
]
```

GET - Récupérer un animal par ID

- **Description** : Renvoie un animal spécifique en fonction de son ID.
- **URL** : `/pets/{id}`
- **Exemple d'utilisation** :

```
curl -X GET http://localhost:8084/pets/1
```

- **Réponse attendue** :

```
{
  "id": 1,
  "name": "Rex",
  "age": 5,
  "owner": "Jack"
}
```

GET - Récupérer les animaux d'un propriétaire

- **Description** : Liste tous les animaux appartenant à un propriétaire donné.
- **URL** : `/pets/owner/{owner}`
- **Exemple d'utilisation** :

```
curl -X GET http://localhost:8084/pets/owner/Peter
```

- **Réponse attendue** :

```
[  
  {  
    "id": 3,  
    "name": "Pepette",  
    "age": 8,  
    "owner": "Peter"  
  },  
  ...  
]
```

POST - Ajouter un nouvel animal

- **Description** : Ajoute un animal dans la base de données.
- **URL** : `/pets/add`
- **Exemple d'utilisation** :

```
curl -X POST http://localhost:8084/pets/add \  
-H "Content-Type: application/json" \  
-d '{  
  "id": 7,
```

```
"name": "Buddy",  
"age": 4,  
"owner": "Alice"  
'}
```

- **Réponse attendue :**

```
{  
  "id": 7,  
  "name": "Buddy",  
  "age": 4,  
  "owner": "Alice"  
}
```

PUT - Modifier un animal existant

- **Description :** Met à jour les informations d'un animal existant en fonction de son ID.
- **URL :** `/pets/update/{id}`
- **Exemple d'utilisation :**

```
curl -X PUT http://localhost:8084/pets/update/1 \  
-H "Content-Type: application/json" \  
-d '{  
  "id": 1,  
  "name": "Milou",  
  "age": 12,  
  "owner": "Peter"  
}'
```

- **Réponse attendue :**

```
{
  "id": 1,
  "name": "Milou",
  "age": 12,
  "owner": "Peter"
}
```

DELETE - Supprimer un animal

- **Description :** Supprime un animal de la base de données via son ID.
- **URL :** `/pets/remove/{id}`
- **Exemple d'utilisation :**

```
curl -X DELETE http://localhost:8084/pets/remove/1
```

- **Réponse attendue :**
 - **Code HTTP :** `204 No Content` (succès sans contenu).

Résumé des points d'accès

Méthode	URL	Description
GET	<code>/commands/ping</code>	Vérifie la connectivité de l'API.
GET	<code>/pets</code>	Liste tous les animaux.
GET	<code>/pets/{id}</code>	Récupère un animal par ID.
GET	<code>/pets/owner/{owner}</code>	Récupère les animaux d'un propriétaire.
POST	<code>/pets/add</code>	Ajoute un nouvel animal.
PUT	<code>/pets/update/{id}</code>	Modifie un animal existant.

DELETE	<code>/pets/remove/{id}</code>	Supprime un animal via son ID.
--------	--------------------------------	--------------------------------

Cette documentation peut être utilisée pour tester et intégrer l'API dans vos applications ou outils de développement (par exemple, Postman).