

## Dierentuin API Documentatie

Gemaakt door: Debo, Deo en Eric

## Inleiding

Deze API geeft je toegang tot alle data van de diertuin, zoals:

1. **Dieren**
2. **Categorieën** (bijvoorbeeld “Zoogdieren”, “Vogels”, etc.)
3. **Verblijven** (Engels: Enclosures)
4. **Dierentuinen** (als je er meer dan één hebt)

Alle data wordt verstuurd en ontvangen in **JSON-formaat**. Bij het bevragen van de API geldt:

- **GET** voor het ophalen (read)
- **POST** voor het toevoegen (create)
- **PUT** voor het bijwerken (update)
- **DELETE** voor het verwijderen (delete)

Als een resource niet bestaat (bijv. je vraagt een dier op met een ID dat niet in de database voorkomt), dan krijg je de status **404 Not Found**.

## De basis-url

**“api/AnimalApi”**

Hiermee beheer je de **dieren** in de diertuin.

## Belangrijke Endpoints

1. **GET /api/AnimalApi?search={zoekterm}**
  - Haal een lijst met dieren op.
  - Met de optie **search** kun je filteren op naam, soort, categorie of verblijf.
  - Voorbeeld: GET /api/AnimalApi?search=Leeuw
2. **GET /api/AnimalApi/{id}**
  - Haal de details op van een specifiek dier (met ID).
  - Voorbeeld: GET /api/AnimalApi/5
3. **POST /api/AnimalApi**
  - Maak een nieuw dier aan.
  - Verstuur de gegevens in de body als JSON.

### Voorbeeld-JSON:

**json**

```
{  
  "Name": "Leeuw",  
  "Species": "Panthera leo",  
  "CategoryId": 1,  
  "Size": 50,  
  "DietaryClass": "Carnivore",  
  "ActivityPattern": "Diurnal",  
  "SpaceRequirement": 10,  
  "SecurityRequirement": "Medium"  
}
```

#### 4. PUT /api/AnimalApi/{id}

- Werk een bestaand dier bij.
- Het ID in de URL moet overeenkomen met het ID in de JSON.
- Voorbeeld: PUT /api/AnimalApi/5

#### 5. DELETE /api/AnimalApi/{id}

- Verwijder een dier.
- Voorbeeld: DELETE /api/AnimalApi/5

### Extra Acties (Sunrise, Sunset, etc.)

#### • GET /api/AnimalApi/{id}/sunrise

- Pas het slaap-/waakgedrag aan voor één dier bij zonsopgang.
- Diurnaal (dagdier) wordt wakker, Nocturnaal (nachtdier) gaat slapen, Cathemeraal (flexibel) wordt wakker.

#### • GET /api/AnimalApi/{id}/sunset

- Pas het gedrag aan bij zonsondergang.
- Diurnaal gaat slapen, Nocturnaal wordt wakker, Cathemeraal gaat slapen.

#### • GET /api/AnimalApi/{id}/feedingtime

- Als het dier een carnivoor is en in een verblijf zit, eet het alle kleinere dieren in dat verblijf.
- Voorbeeld: GET /api/AnimalApi/5/feedingtime

#### • GET /api/AnimalApi/{id}/checkconstraints

- Controleer of een dier genoeg ruimte heeft en of de beveiliging van het verblijf voldoende is.

## 2. CategoryApiController

### Basis-URL:

api/CategoryApi

Hiermee beheer je de **categorieën** waarin dieren kunnen vallen (bijvoorbeeld “Zoogdieren”, “Vogels”, etc.).

### Belangrijke Endpoints

#### 1. GET /api/CategoryApi?search={zoekterm}

- Haal een lijst met categorieën op.
- Eventueel gefilterd op naam.

#### 2. GET /api/CategoryApi/{id}

- Haal de details op van één categorie (met ID).

#### 3. POST /api/CategoryApi

- Voeg een nieuwe categorie toe.
- Voorbeeld-JSON:

json

```
{  
  "Name": "Vissen"  
}
```

#### 4. PUT /api/CategoryApi/{id}

- Werk een bestaande categorie bij.
- Het ID moet overeenkomen.

#### 5. DELETE /api/CategoryApi/{id}

- Verwijder een categorie.

### 3. EnclosureApiController

#### Basis-URL:

api/EnclosureApi

Deze controller beheert de **verblijven** (enclosures) in de diertuin.

#### Belangrijke Endpoints

1. **GET /api/EnclosureApi?search={zoekterm}**

- Haal een lijst met verblijven op, eventueel gefilterd op naam.
- Voorbeeld: GET /api/EnclosureApi?search=Savannah

2. **GET /api/EnclosureApi/{id}**

- Haal de details op van één specifiek verblijf.

3. **POST /api/EnclosureApi**

- Maak een nieuw verblijf aan.
- Voorbeeld-JSON:

json

```
{  
  "Name": "Nieuw Verblijf",  
  "Climate": "Temperate",  
  "HabitatType": "Grassland",  
  "SecurityLevel": "Medium",  
  "Size": 200  
}
```

4. **PUT /api/EnclosureApi/{id}**

- Werk een bestaand verblijf bij.
- ID moet overeenkomen.

5. **DELETE /api/EnclosureApi/{id}**

- Verwijder een verblijf.

### Extra Acties (Sunrise, Sunset, enz.)

- **GET /api/EnclosureApi/{id}/sunrise**
  - Zet alle dieren in dat verblijf op wakker/slaper, afhankelijk van hun activiteitspatroon (maar dan voor zonsopgang).
- **GET /api/EnclosureApi/{id}/sunset**
  - Zelfde idee, maar dan voor zonsondergang.
- **GET /api/EnclosureApi/{id}/feedingtime**
  - Voer de “feedingtime”-actie uit in dit verblijf. Carnivoren eten alle kleinere dieren.
- **GET /api/EnclosureApi/{id}/checkconstraints**
  - Controleer of er genoeg ruimte is en of de beveiliging voldoende is voor alle dieren in dit verblijf.

### 4. ZooApiController

#### Basis-URL:

api/ZooApi

Deze controller is voor het beheren van de **dierentuinen** zelf, mocht je er meerdere hebben.

#### Belangrijke Endpoints

1. **GET /api/ZooApi**
  - Haal een lijst op met alle diertuinen, inclusief hun verblijven en dieren.
2. **GET /api/ZooApi/{id}**
  - Haal de details op van één diertuin (met ID).
3. **POST /api/ZooApi**
  - Maak een nieuwe diertuin aan.
  - Voorbeeld-JSON:

**json**

```
{  
  "Name": "Nieuwe Dierentuin"  
}
```

#### 4. **PUT /api/ZooApi/{id}**

- Werk een bestaande diertuin bij.

#### 5. **DELETE /api/ZooApi/{id}**

- Verwijder een diertuin.

### **Extra Acties**

- **GET /api/ZooApi/{id}/sunrise**

- Voer bij elke enclosure in de diertuin de sunrise-actie uit.
- Hierdoor worden alle dieren aangepast op basis van hun activiteitspatroon (diurnaal wordt wakker, nocturnaal gaat slapen, etc.).

- **GET /api/ZooApi/{id}/sunset**

- Zelfde idee als sunrise, maar dan zonsondergang-logica.

- **GET /api/ZooApi/{id}/feedingtime**

- Alle verblijven in deze diertuin voeren feedingtime uit (carnivoren eten kleinere dieren, etc.).

- **GET /api/ZooApi/{id}/checkconstraints**

- Controleer in elke enclosure of er genoeg ruimte is en of de beveiliging voldoende is.

- **GET /api/ZooApi/{id}/autoassign?removeExisting={true/false}**

- Wijs niet-toegewezen dieren automatisch toe aan (nieuwe) verblijven.
- Als **removeExisting** = true, dan verwijdert de API eerst de bestaande verblijven en maakt er een nieuw verblijf bij.