

# Installatiehandleiding

Backend systeem voor een dierenarts

Auteur: **Tanja van Hunen**

Datum: **Februari 2022**

# Inleiding

Dit document behoort samen met een verantwoordingsdocument, technisch ontwerp en source code tot een geheel voor de afronding van het vak 'backend' aan Hogeschool Novi te Utrecht. In dit document wordt stapsgewijs beschreven hoe de applicatie geïnstalleerd moet worden.

# Inhoudsopgave

[Inleiding](#)

[Inhoudsopgave](#)

[1. Benodigdheden](#)

[2. Clone git repository](#)

[3. Database](#)

[4. Demo data](#)

[5. Applicatie starten](#)

[6. Postman](#)

# 1. Benodigdheden

Om de applicatie te kunnen starten heb je een paar benodigdheden nodig. Namelijk het volgende:

- IntelliJ IDEA ( hier te downloaden: [Intelij download pagina](#) )
- Docker-desktop ( hier te downloaden: [Docker-desktop download pagina](#) )
- Postman ( hier te downloaden: [Postman download pagina](#) )
- JDK 14 ( hier te downloaden: [JDK 14 download pagina](#) )

Voor alle applicaties die je nog niet hebt moet je na het downloaden ook de installaties hiervan doen. Dit wijst zichzelf omdat alle download bestanden een installer hebben.

## 2. Clone git repository

Om te beginnen clone je de repository van github.

1. Ga met je terminal naar de map waar je deze bestanden neer wilt zetten. Als je niet weet hoe dit moet bekijk dan de volgende link voor windows: [Basic cmd use](#) of de volgende link voor mac en linux: [Basic terminal use](#)
2. Vervolgens ga je het project clonen van github. Dit doe je met het volgende commando:  
git clone <https://github.com/EmrysAvalon/legendary-octo-spork.git>

## 3. Database

Om de applicatie te kunnen starten moet eerst de database aan staan omdat je anders tegen fouten aanloopt. Controleer of je docker-desktop hebt geïnstalleerd bij stap 1. Vervolgens moet je controleren of de poort 5432 vrij is op je machine. Als dit niet het geval is maak deze dan vrij door het draaiende programma op die poort af te sluiten.

Als de poort vrij is en docker-desktop is geïnstalleerd voer dan het volgende commando uit in de terminal of cmd:

```
docker run --name postgres-tanja -p 5432:5432 -e POSTGRES_PASSWORD=springboot -e POSTGRES_USER=springboot -e POSTGRES_DB=springboot -d postgres
```

Het commando werkt als volgt:

– name postgres-tanja (geeft de naam postgres-tanja aan de docker container waardoor deze makkelijk te identificeren is als je de database opnieuw wilt starten bijvoorbeeld na een herstart van de computer. Ook als je de applicatie niet meer wilt gebruiken is deze makkelijk te identificeren en verwijderen.)

-p 5432:5432 (geeft aan dat de docker container moet draaien op poort 5432.)

- e POSTGRES\_PASSWORD=springboot (Geeft het wachtwoord op van de postgres database.)
- e POSTGRES\_USER=springboot (Geeft de gebruikersnaam op van de postgres database.)
- e POSTGRES\_DB=springboot (Geeft de naam van de postgres database.)

Als je iets aanpast aan het commando betekent dit ook dat je in het application.properties bestand van het project deze gegevens moet aanpassen.

Om de database gegevens te zien na het draaien van de applicatie kan je de database toevoegen in IntelliJ bij het database tabje. Hierin vul je de gegevens zoals de gebruikersnaam, wachtwoord, database naam, poort en localhost mee die met dit commando zijn ingesteld.

## 4. Demo data

Voor het gebruik van de applicatie is wat data nodig. Om tijd te besparen en gelijk aan de slag te kunnen gaan is er wat demo data gemaakt. Deze gegevens kan je terug vinden in het data.sql bestand. In dit bestand zijn de gebruikers wachtwoorden encrypted meegegeven. Ik heb voor alle beschikbaren rollen een gebruiker aangemaakt.

Voor een receptionist zijn hier de volgende inloggegevens:

gebruikersnaam: diana.slater

wachtwoord: wBb7n4C!

Voor een administratief medewerker zijn hier de volgende inloggegevens:

gebruikersnaam: irene.wright

wachtwoord: oF6F\_@my

Voor een administrator zijn hier de volgende inloggegevens:

gebruikersnaam: connor.butler

wachtwoord: 9PZPP@Gt

Omdat er een extra gebruiker aangemaakt kan worden heb ik een extra set gebruikersgegevens. Deze staat nog niet in de applicatie maar maakt het makkelijker om te testen via postman. De inloggegevens van deze extra gebruiker zijn:

gebruikersnaam: penelope.sanderson

wachtwoord: yTVFB!T3

Omdat er bij het aanmaken van een nieuwe gebruiker een gebruikersrol moet worden meegegeven hierbij alle mogelijke gebruikersrollen voor het overzicht:

ROLE\_RECEPTIONIST

ROLE\_ADMINISTRATIVE\_WORKER

ROLE\_ADMIN

## 5. Applicatie starten

Om de applicatie te starten open je de applicatie IntelliJ. Hierin open je vervolgens het project wat je net gecloned hebt tijdens stap 2.

Zoek het bestand `DierenartsApplication`. Deze staat hier  
→ `src/main/java/com/example/dierenarts/DierenartsApplication.java`

Wacht tot IntelliJ klaar is met het inladen van alle gegevens dit kan je onderaan de balk zien. Als daar nog een balkje wordt geladen moet je nog even wachten. Als dit klaar is zal er ook een afspeelknop verschijnen links naast de tekst `public class DierenartsApplication`.

Klik op de afspeelknop en wacht tot de applicatie is gestart.

## 6. Postman

Om de applicatie te testen kan je gebruik maken van postman. In de documentatiemap staat een bestand dat heet `"Dierenarts.postman-collection.json"` dit bestand kan je importeren in postman.

Hierdoor staan alle gegevens goed zoals de authenticatie. Mocht dit niet het geval zijn moet je even op Auth drukken, basic authentication selecteren en de juiste gebruikersgegevens invullen. De mogelijke acties staan verdeeld in mappen zoals de gebruikersrollen deze kunnen uitvoeren.

Voor het uploaden van bestanden moet je nog wel even een bestand mee geven in de header. Voor de bestanden is een limiet van 2MB.

Houd er rekening mee dat elke keer als de applicatie wordt herstart de gegevens weer terug gaan zoals ze in `data.sql` staan.