

Verantwoordingsdocument bij PlantjesPlanner

Eindopdracht NOVI Bootcamp Full Stack Developer

Inhoudsopgave

Inhoudsopgave	2
Inleiding	2
Benodigdheden	3
Installatie van de applicatie	4
De PlantjesPlanner starten	5
Testgebruikers	5
Endpoints	6

Inleiding

Met de PlantjesPlanner kan informatie over tuinplanten worden verzameld en beheerd. Gebruikers kunnen een account aanmaken om planten te vergelijken en uit te kiezen voor een tuin of balkon. Ook kan je gegevens vastleggen over de tuin om de planten te kunnen vergelijken met de omstandigheden van de standplaats.

Deze Installatiehandleiding beschrijft de benodigdheden om de applicatie “PlantjesPlanner” te installeren en te starten. De applicatie bestaat uit drie onderdelen:

- Frontend - React
- Backend - Java / Spring Boot
- Database - PostgreSQL

Om de backend los van de frontend te kunnen testen, staan in het hoofdstuk “Endpoints” tabellen met een beschrijving van de REST endpoints met daarbij voorbeelden voor JSON-requests om te kopiëren en plakken in bijvoorbeeld Postman.

Deze handleiding is bedoeld voor ontwikkelaars die beschikken over een Java- en JavaScript-ontwikkelomgeving (hier wordt uitgegaan van IntelliJ en WebStorm), maar geen ervaring hebben met Spring Boot, React of PostgreSQL.

Benodigdheden

Het is mogelijk om alle onderdelen zonder ontwikkelomgevingen te gebruiken te starten maar dit vergt zoveel (foutgevoelige) handelingen die ook nog verschillen per operating system, dat deze handleiding uitgaat van gebruik met behulp van ontwikkelomgevingen voor de frontend, backend en database. In onderstaande tabel staan de benodigdheden om de PlantjesPlanner te installeren en starten. Behalve de IntelliJ IDEA en Webstorm is alles gratis te

benodigde software	link naar download	beschrijving
IntelliJ IDEA	https://www.jetbrains.com/idea/download/	IDE voor programmeren in o.a. Java, wordt gebruikt om de backend applicatie te starten
Java JDK 21	https://www.oracle.com/java/technologies/downloads/#java21	Development kit voor Java 21: dient om de backend applicatie te draaien.
Webstorm	https://www.jetbrains.com/webstorm/download/	IDE voor programmeren in o.a. JavaScript, wordt gebruikt om de frontend applicatie te starten
Node.js + NPM	https://nodejs.org/en/download	Node.js is een runtime omgeving voor JavaScript. NPM wordt gebruikt om JavaScript dependencies te installeren
PostgreSQL	https://www.postgresql.org/download/	relationele database
pgAdmin	https://www.pgadmin.org/download/	starten en beheren van de database

downloaden van internet. Volg de instructies op de websites om de software te installeren.

Een gratis alternatief voor JetBrains' IntelliJ en Webstorm is bijvoorbeeld Microsoft Visual Studio Code (<https://code.visualstudio.com/download>).

projectnaam	URL naar Github pagina	beschrijving
plantjesplanner-be	https://github.com/szuideweg/plantjesplanner-be	backend applicatie
plantjesplanner-fe	https://github.com/szuideweg/plantjesplanner-fe	frontend applicatie

Installatie van de applicatie

Hier staat stap voor stap beschreven waar de PlantjesPlanner te downloaden is en hoe te installeren en configureren

1. **download en installeer de benodigde software** uit het vorige hoofdstuk indien nodig. Volg de instructies op de downloadpagina's
2. **PlantjesPlanner downloaden van Github.** Het bestaat uit twee losse projecten. Download ze als .zip en pak uit op een maplocatie naar keuze.
3. **Backend instellen en controleren:** Start IntelliJ en open het project "plantjesplanner-be". controleer of Java 21 is ingesteld als Project SDK door in de geïntegreerde terminal in te voeren: `java -version` . Zo niet, ga naar file > project structure > project settings . Selecteer bij "SDK" een Java 21 versie.
4. **afbeeldingen bij testdata klaarzetten:** kopieer of verplaats de map `be_plantjesplanner/plant_avatars` met daarin de afbeeldingen bij de testdata een niveau hoger zodat het in dezelfde map als het project `be_plantjesplanner` staat. Een alternatieve manier: als de map tijdens het opstarten van de backend niet bestaat wordt deze automatisch aangemaakt. Je moet dan zelf de afbeeldingen naar de nieuwe map kopiëren.
5. **database aanmaken en instellen**

1. start pgAdmin
2. ga in de navigatiebalk naar servers > PostgreSQL . Rechtermuisklik > Create > Database.
3. In het tabblad General bij Database, noem de database: `plantjesplanner`
4. Rechtermuisklik op de nieuwe database "plantjesplanner"> open de Query tool
5. Kopieer en plak onderstaande query om een nieuwe database user aan te maken voor de backend applicatie om te verbinden met de database:

```
CREATE USER springboot WITH PASSWORD 'springboot';  
GRANT ALL PRIVILEGES ON DATABASE plantjesplanner TO springboot;
```

6. controleer dat de database verbinding heeft met de databaseserver: het icoontje van de database niet meer grijs met een rood kruisje maar geel gekleurd.

6. Frontend Installeren

1. Open Webstorm en de geïntegreerde Terminal en controleer of Node.js en NPM goed zijn geïnstalleerd met de commando's: `node -v` en `npm -v` . Als beiden een versienummer tonen was de installatie in het vorige hoofdstuk geslaagd.

2. Node.js koppelen aan Webstorm: Ga naar Settings in het hoofdmenu > Languages & Frameworks > Node.js . Bij Node Interpreter, klik add. Kies Local en selecteer het Node.js pad. Klik Apply en OK.
3. Installeer dependencies: Voer deze commando's uit in de Terminal van Webstorm: *npm install*

De PlantjesPlanner starten

Wanneer de benodigde software, de frontend en backend zijn geïnstalleerd en de database is aangemaakt, kan de applicatie gestart worden:

1. Open pgAdmin. Klik op de plantjesplanner database en controleer dat deze verbonden is met de database server: het database icoontje is geel.
2. Open IntelliJ IDEA en open het project be_plantjesplanner. Navigeer naar de main klasse: `../be_plantjesplanner/src/main/java/nl/novi/be_plantjesplanner/BePlantjesplannerApplication.java` .Rechtermuis op dit bestand > Run `BePlantjesplannerApplication` .
3. Open Webstorm en open het project fe_plantjesplanner. Ga naar de Terminal van Webstorm en voer in: *npm run dev*

Testgebruikers

Hieronder staan de credentials van de twee testgebruikers uit de test data.

omschrijving	username	password	role
gebruiker met designer-rol	designer_test	geheim	Designer
gebruiker met admin-rol	admin_test	super_geheim	Admin

Er zijn twee rollen: Designer en Admin. Iedereen kan een nieuwe Designer aanmaken, maar alleen een Admin kan een andere Admin aanmaken. In het volgende hoofdstuk staat per endpoint genoemd welke rollen er toegang tot hebben.

Endpoints

De tabellen in dit hoofdstuk beschrijven de REST-endpoints van de PlantjesPlanner backend met voorbeelden. Ze zijn zo opgesteld dat je de voorbeelden rechtstreeks kunnen kopiëren en plakken in Postman. Voor de instellingen die niet expliciet zijn genoemd in de tabellen worden de standaard waarden van Postman gebruikt.

De meeste endpoints zijn beveiligd met JWT Bearer Authentication. Elk beschermd endpoint vereist een geldig JWT dat met de Authorization header wordt meegestuurd. Een JWT kan worden opgehaald via het /login endpoint met geldige inloggegevens in de request body. Zie de tabel “JWT ophalen” voor informatie over het request.

De response is een token code. Het makkelijkst is om Postman automatisch de header in te laten vullen:

1. Ga naar het tabblad “Auth” van het Postman request
2. Selecteer bij Auth Type: Bearer token
3. kopieer in het veld “token” de code uit de response body van het login request.
4. De authorization header is nu toegevoegd aan het HTTP request.

/login - JWT bearer token ophalen

JWT ophalen (Inloggen) - publiek

	waarde	voorbeeld
HTTP method	POST	
URL	http://localhost:8080/login	http://localhost:8080/login
Authentication	no authentication	
Body settings	raw JSON	
Body content	{ "username": " ", "password": " " }	{ "username": "designer_test", "password": "geheim" }

/users - gebruikersbeheer

nieuwe gebruiker met Designer-rol - publiek

	waarde	voorbeeld
HTTP method	POST	
URL	http://localhost:8080/users/register	http://localhost:8080/users/register
Authentication	no authentication	
Body settings	raw JSON	
Body content	{ "username": " ", "password": " " }	{ "username": "test123", "password": "blijje_plant" }

nieuwe gebruiker met Admin-rol - admin

	waarde	voorbeeld
HTTP method	POST	
URL	http://localhost:8080/users/admin	http://localhost:8080/users/admin
Authentication	no authentication	
Body settings	raw JSON	
Body content	{ "username": " ", "password": " " }	{ "username": "marietje", "password": "blijje_plant" }

eigen gebruikersinformatie ophalen - designer & admin

	waarde	voorbeeld
HTTP method	GET	
URL	http://localhost:8080/users/me	http://localhost:8080/users/me
Authentication	jwt bearer token	
Body settings	none	
Body content		

/plants - planten beheren

nieuwe plant aanmaken met afbeelding - admin

	waarde	voorbeeld
HTTP method	POST	
URL	http://localhost:8080/plants	http://localhost:8080/plants
Authentication	jwt bearer token	
form data plant	key (text): plant content type: application/JSON	
value "plant"	{ "dutchName": " ", "latinName": " ", "description": " ", "height": "", "footprint": " ", "bloomColorHex": " ", "bloomColorGroup": "", "published": , "localeDto": { "sunlight": "", "moisture": "", "windTolerance": "", "soilType": "", "openGroundOnly": "" }, "bloomingCalendarDto": { "january": , "february": "march":, "april":, "may":, "june":, "july":, "august":, "september":, "october": , "november": , "december": } }	{ "dutchName": "Schuimbloem" , "latinName": "Tiarella cordifolia", "description": "Fijne laagblijvende bodembedekker voor een schaduwrijke plek. Blijft ook nog eens groen in de winter. ", "height": "0.25", "footprint": "0.1", "bloomColorHex": "#FFE4E1", "bloomColorGroup": "ROZE", "published": true, "localeDto": { "sunlight": "HALFSCHADUW", "moisture": "VOCHTIG", "windTolerance": "GEMIDDELD", "soilType": "geen voorkeur", "openGroundOnly": "false" }, "bloomingCalendarDto": { "january": false, "february": false, "march": false, "april": true, "may": true, "june": true, "july": true, "august": false, "september": false, "october": false, "november": false, "december": false} }
form data file	key(file): image content type: auto	
value "image"	een afbeeldingsbestand	

let op: POST plant gebruikt geen raw JSON body, maar form data met twee onderdelen: De plant gegevens in minified JSON en een afbeeldingsbestand in image.

plant bewerken- admin

	waarde	voorbeeld
HTTP method	PUT	
URL	http://localhost:8080/plants/{plantId}	http://localhost:8080/plants/2
Authentication	jwt bearer token	
form data plant	key (text): plant content type: application/JSON	
value "plant"	<pre>{ "dutchName": " ", "latinName": " ", "description": " ", "height": "", "footprint": " ", "bloomColorHex": " ", "bloomColorGroup": "", "published": , "localeDto": { "sunlight": "", "moisture": "", "windTolerance": "", "soilType": "", "openGroundOnly": "" }, "bloomingCalendarDto": { "january": , "february": "march":, "april":, "may":, "june":, "july":, "august":, "september":, "october": , "november":, "december": } }</pre>	
form data file	key(file): image content type: auto	
value "image"	een afbeeldingsbestand	

let op: PUT plant gebruikt geen raw JSON body, maar form data met twee onderdelen: De plant gegevens in minified JSON en een afbeeldingsbestand in image.

alle planten in catalogus ophalen - admin & designer

	waarde	voorbeeld
HTTP method	GET	
URL	http://localhost:8080/plants	http://localhost:8080/plants
Authentication	jwt bearer token	
Body settings	none	
Body content		

plant ophalen op id-nummer - admin & designer

	waarde	voorbeeld
HTTP method	GET	
URL	http://localhost:8080/plants/{id}	http://localhost:8080/plants/2
Authentication	jwt bearer token	
Body settings	none	
Body content		

plant zoeken op Nederlandse naam - admin & designer

	waarde	voorbeeld
HTTP method	GET	
URL	http://localhost:8080/plants/search?name={plant.dutchName}	http://localhost:8080/plants/search?name=hosta
Authentication	jwt bearer token	
Body settings	none	
Body content		

afbeelding van plant ophalen op id-nummer van plant — admin & designer

	waarde	voorbeeld
HTTP method	GET	
URL	http://localhost:8080/plants/{id}/avatar	http://localhost:8080/plants/1/avatar
Authentication	jwt bearer token	
Body settings	none	
Body content		

plant verwijderen - admin

	waarde	voorbeeld
HTTP method	DELETE	
URL	http://localhost:8080/plants/{id}	http://localhost:8080/plants/2
Authentication	jwt bearer token	
Body settings	none	
Body content		

/Designs - tuinontwerpen

eigen ontwerp ophalen - designer

	waarde	voorbeeld
HTTP method	GET	
URL	http://localhost:8080/designs/me	http://localhost:8080/designs/me
Authentication	jwt bearer token	
Body settings	none	
Body content		

eigen ontwerp bewerken - designer

	waarde	voorbeeld
HTTP method	PUT	
URL	http://localhost:8080/designs/me	http://localhost:8080/designs/me
Authentication	jwt bearer token	
Body settings	raw JSON	
Body content	<pre>{ "title": "", "gardenSize": , "localeDto": { "sunlight": "", "moisture": "", "windTolerance": "", "soilType": "", "openGroundOnly": } }</pre>	<pre>{ "title": "Mijn prachtige Achtertuin", "gardenSize": 40.0, "localeDto": { "sunlight": "ZONNIG", "moisture": "MATIG_VOCHTIG", "windTolerance": "GEMIDDELD", "soilType": "zand", "openGroundOnly": true } }</pre>

/selected-plants - planten aan ontwerp koppelen

plant toevoegen aan eigen ontwerp - designer

	waarde	voorbeeld
HTTP method	POST	
URL	http://localhost:8080/designs/me/plants	http://localhost:8080/designs/me/plants
Authentication	jwt bearer token	
Body settings	raw JSON	
Body content	{ "plantId": "amount": }	{ "plantId":2, "amount":5 }

aantal van een plant bij eigen ontwerp wijzigen - designer

	waarde	voorbeeld
HTTP method	PATCH	
URL	http://localhost:8080/designs/me/plants/2?amount={amount}	http://localhost:8080/designs/me/plants/2?amount=16
Authentication	jwt bearer token	
Body settings	none	
Body content		

plant verwijderen bij eigen ontwerp - designer

	waarde	voorbeeld
HTTP method	DELETE	
URL	http://localhost:8080/designs/me/plants/{selectedPlantId}	http://localhost:8080/designs/me/plants/2
Authentication	jwt bearer token	
Body settings	none	
Body content		