Dependencies & Plugins

Inhoud

[Inleiding 1](#_Toc156220951)

[Back-end 2](#_Toc156220952)

[Spring security 2](#_Toc156220953)

[Modelmapper 2](#_Toc156220954)

[Spring start mail 2](#_Toc156220955)

[MariaDB java client 2](#_Toc156220956)

[H2 2](#_Toc156220957)

[Springboot starter web 2](#_Toc156220958)

[Spring boot starter test 2](#_Toc156220959)

[Spring boot starter data jpa 2](#_Toc156220960)

[Lombok 2](#_Toc156220961)

[Front-end 3](#_Toc156220962)

[React 3](#_Toc156220963)

[Axios 3](#_Toc156220964)

[Bootstrap 3](#_Toc156220965)

[Heatmap.js 3](#_Toc156220966)

[Recharts 3](#_Toc156220967)

[Leaflet 3](#_Toc156220968)

[Algemeen 4](#_Toc156220969)

[MariaDB 4](#_Toc156220970)

[Meet je stad API 4](#_Toc156220971)

# Inleiding

In dit document bespreken we alle dependencies van het huidige project. Hierbij bespreken we wat de dependency doet en waarom deze in het project zit. Dit splitsen we op in 3 stukken: back-end, front-end en algemeen.

# Back-end

Hier wordt behandeld welke externe software wij gebruiken voor ons Java backend. Voor ieder onderdeel wordt de naam gegeven en beschreven wat het doet en waarvoor het gebruikt wordt.

## Spring security

Deze software is de standaard voor security in spring boot. Deze kan gebruikt worden om de endpoints van een API te vergrendelen. Deze kan zo gebruikt worden voor authenticatie en autorisatie om zo te bepalen of iemand bij een endpoint mag. Op het moment wordt deze nog niet gebruikt. Wel is de bedoeling dat een account functionaliteit deze mogelijk gaat gebruiken. Vandaar dat deze nog in het project zit.

## Modelmapper

Dit is een stuk software die Java objecten omzet naar JSONs. Deze wordt gebruikt door ongebruikte niet werkende code in de UserService. Wanneer deze mogelijk vervalt door het vervangen met een werkende feature zou het mogelijk moeten zijn om de modelmapper uit het project te halen.

## Spring start mail

Verstuurd geautomatiseerd e-mails. Deze wordt net als modelmapper gebruikt voor de oude user features. Hier is ook de kans dat deze nog gebruikt gaat worden in de uitwerking van de aankomende user features.

## MariaDB java client

Is de software die het JPA ORM nodig heeft om te communiceren met de mariaDB. Dit wordt voor een aantal features in de data caching gebruikt branch. In de dev branch wordt deze alleen gebruikt voor wijken.

## H2

Verzorgt in memory databases. Dit betekent dat het een database aanmaakt voor tijdelijk gebruik die aan het einde van de sessie wordt verwijderd. Deze zou gebruikt moeten worden in integratie testen, maar dit gebeurt nog niet. Deze staat er dus alleen in voor toekomstig gebruik.

## Springboot starter web

Levert de essentiële onderdelen voor het simpel bouwen van REST API’s. Deze wordt in alle features gebruikt.

## Spring boot starter test

Is voor het maken van unit testen. Deze wordt nu voor de unit testen gebruikt.

## Spring boot starter data jpa

Deze dependency is een Spring extensie op de Java Persistence API (JPA), het wordt gebruikt in Object Relational Mapping (ORM) om de conversie tussen tabellen en objecten te voorzien. Momenteel wordt deze alleen gebruikt voor het ophalen van wijken uit de database, maar wanneer data caching geïmplementeerd wordt zal deze in alle GET requests gebruikt worden.

## Lombok

Levert de optie om met annotaties code te versimpelen. Zo kunnen met annotaties bijvoorbeeld constructors, getters en setters gegenereerd worden. Dit wordt in bijna iedere klasse gebruikt.

# Front-end

Hier wordt behandeld welke externe software wij gebruiken voor ons Javascript front-end. Voor ieder onderdeel wordt de naam gegeven en beschreven wat het doet en waarvoor het gebruikt wordt.

## React

Is de hoofd-library van de applicatie. Deze verzorgt de basis voor de componenten en anderen features voor het maken van een dynamische webapplicatie. Vandaar wordt deze ook gebruikt in ieder deel van de applicatie.

## Axios

Levert de mogelijkheid om simpel en netjes calls te doen naar een API. Vandaar wordt deze weer voor alles gebruikt dat informatie op moet halen.

## Bootstrap

Is een library die opmaak voor standaard componenten levert. Deze wordt dus gebruikt voor het netjes maken van het user interface.

## Heatmap.js

Zorgt voor de huidige heat map. Deze wordt dus ook gebruikt voor het weergeven van de heatmap van alle data. Deze kan komen te vervallen mocht de heatmap verder ontwikkeld worden.

## Recharts

Is de library voor de grafieken. Deze worden gebruikt voor het weergeven van de geschiedenis van metingen.

## Leaflet

Levert de kaart en het coördinatensysteem. Daarnaast horen de pop-ups en markers ook bij deze library. Dit wordt dus gebruikt voor het weergeven van de meetpunten, wijken en heatmap op de kaart. Plus de pop-ups waar de grafieken van de geschiedenis in zitten.

# Algemeen

Hier wordt behandeld welke externe technieken het hele systeem afhankelijk van is.

## MariaDB

Is het type database dat gebruikt wordt om de wijken op te slaan. In de data caching worden hier ook de metingen en meetstations opgeslagen. De database kan relatief makkelijk vervangen worden. Dit komt doordat de huidige back-end de enige is die met de database te maken heeft.

## Meet je stad API

Levert de metingen die gebruikt worden in de applicatie. Deze wordt bijgehouden door het overkoepelend meet je stad project waar deze website aan bijdraagt. Vandaar zal het systeem hier altijd afhankelijk van blijven.