Front-end in ReactJS

# Inhoud

* ReactJS
* Libraries
* Basis uitleg ReactJS
* Test plan

# ReactJS

* Levert een interactieve website aan de gebruiker
* Alle functies worden hierin gemaakt zoals: Heatmap kunnen zien, stations beheren etc.

De front-end van het project is verantwoordelijk voor het leveren van interactieve componenten aan de gebruiker. Dit is in vorm van webpagina's gemaakt in HTML, CSS en vooral Javascript (ReactJS). In de front-end is er veel data nodig, hiervoor wordt een library genaamd ‘Axios’ gebruikt. Deze library helpt met het maken van REST calls die de back-end kan ontvangen. Eenmaal de data opgehaald is kan het weergegeven worden aan de gebruiker.

# Libraries

De vijf grote libraries die gebruikt worden zijn hieronder beschreven:

**Leaflet**

*Een light-weight instance van openstreetmap die ervoor zorgt dat we een aanpasbare kaart kunnen weergeven*

**Heatmaplayer**

*Een layer bovenop leaflet waarmee we ‘hitte’ velden kunnen weergeven op de kaart op basis van onze gegevens (Voornamelijk de weerstations)*

**Axios**

*Helpt met het maken van REST calls naar de back-end om ervoor te zorgen dat de applicatie aan data kan komen*

**Cypress**

*Een end-to-end test tool die het mogelijk maakt om tests te schrijven die gebruikers input simuleren*

**React-router**

*Een ‘hook’ van react die het mogelijk maakt om component aan routes (endpoints zoals ‘/overons’) te koppelen*

# Basis uitleg ReactJS

Als ReactJS nog niet bekend is is het erg handig om even te kijken naar het volgende gedeelte, hierin wordt ReactJS uitgelegd. *Zie je dit liever in video vorm? Dan is de video hieronder zeer aangeraden*

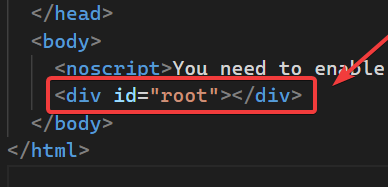
<https://www.youtube.com/watch?v=hQAHSlTtcmY>

In de React applicatie bestaan er veel bestanden, alleen is niet van elk bestand even handig om te onthouden wat ze precies doen. Hieronder is een kort overzicht voor wat welk bestand precies voor dient.

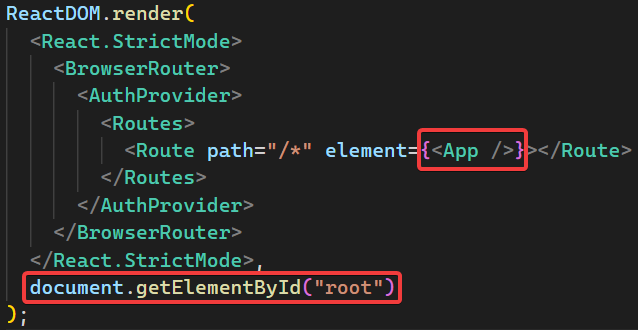


Om een pagina (Oftewel component(s)) in te laden zijn er verschillende bestanden nodig, dit loopt op de volgorde beschreven hieronder:

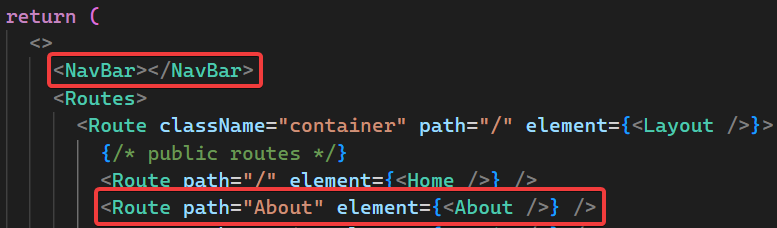
1. ‘public’ > ‘index.html’ Dit is het bestand dat als eerst wordt ingeladen als er naar de basis link wordt genavigeerd. Hierin staat het ‘root’ element waarin later meer ingeladen wordt.



1. ‘src’ > ‘index.js’ Hierin wordt het 'root’ element opgehaald van het ‘index.html’ bestand en vervolgens het volgende bestand in ingeladen

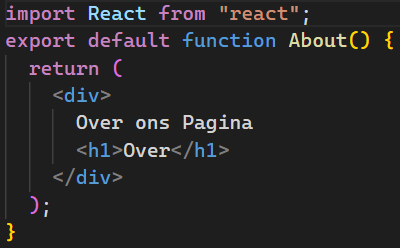


1. ‘src’ > ‘app.js’ Hierin wordt ‘react-router’ gebruikt om ‘/’ (endpoints) te linken aan componenten. Zo wordt altijd het 'navbar’ component in geladen om ervoor te zorgen dat op elk document de navbar te zien is. Hierna wordt er gekeken naar welke endpoint er precies geroepen wordt om vervolgens te bepalen welk component ingeladen moet worden



*Voor demonstratie pakken we een simpele component*

1. 'src’ > ‘components’ > ‘about.js’ Hierin zien we een component die simpele HTML returned. Dit is het einde van de run als het gaat om het inladen van een simpel component



# Test plan

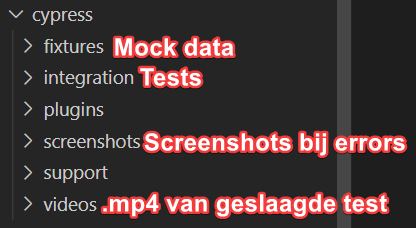
In de front-end staan meerdere test vormen opgesteld, zo is er een unit test en een end-to-end test tool opgezet.

**Unit test:**

De code coverage met de unit tests is op moment van schrijven erg zwak maar wel aanwezig. Door in het project ‘npm run test’ te typen in een terminal wordt de test gedraaid en resultaat weergegeven.

**End-to-end:**

Voor het draaien van end-to-end tests is een library genaamd ‘cypress’ gebruikt. In de ‘cypress’ folder zijn de volgende onderdelen te vinden:



Cypress draait niet terwijl het project draait, deze tool moet apart opgestart worden. Om dit simpel te doen kan je het volgende draaien in een terminal:

npx cypress run – headed

* Run alle tests die geschreven zijn
* --headed zorgt ervoor dat je kan zien wat de test aan het doen is

npx cypress open

* Opent de test tool zodat je handmatig tests kan draaien