

02 – Postcode en Google Maps applicatie

Bouw een applicatie die gebruikmaakt van een Postcode API en Google maps. Met de applicatie kunnen mensen een postcode intypen, waarna detailgegevens van die postcode (straat, stad, provincie) worden getoond. Vervolgens wordt een Google Maps-kaartje geopend waarin de betreffende locatie inclusief een marker wordt getoond.

Je kunt dit opzetten volgens het principe van microservices:

1. Je app praat met jouw (gateway-) server.
2. Server stuurt request naar de API en transformeert/manipuleert desgewenst de resultaten.
3. Server stuurt kant-en-klare gegevens terug naar je app.

De applicatie heeft de volgende requirements:

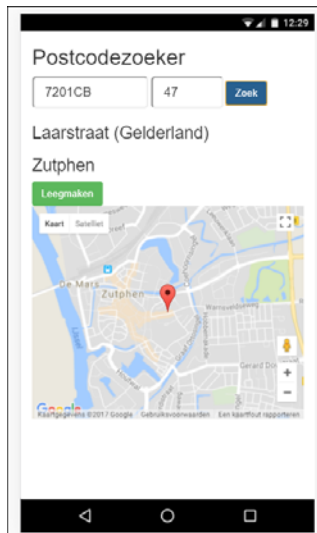
- De app gebruikt een front-end [framework] naar keuze: Angular, Vue , React, Flutter, Nuxt, Nest.js, Vanilla HTML/CSS/JavaScript, of nog anders.
- De architectuur van de app gebruikt het API Gateway-principe. Beide API's *zijn* geschikt om rechtstreeks aan te spreken vanuit je frontend en JSON retour te geven, maar probeer dit op te zetten volgens het in de sessie besproken microservices-principe.
- Zorg voor aantrekkelijke vormgeving. Als je een webapp maakt, mag je een CSS-framework zoals Bootstrap gebruiken, maar dit is niet verplicht. Je mag ook aanvullende libraries als Angular Material of PrimeNG gebruiken. Dit is ook niet verplicht. Bij Flutter/Ionic/... gebruik je natuurlijk de mobiele widgets die voor dat platform beschikbaar zijn.
- De app moet bruikbaar zijn op mobiele apparaten (responsive).
- De app bevat twee zoekvakken (postcode en huisnummer) en een knop *Zoek*.
 - Als de bezoeker een postcode invult en op *Zoek* klikt, wordt het bijbehorende adres, de stad en de provincie getoond.
 - Onder het adres wordt een Google Map van de betreffende locatie getoond.
- Er is een knop *Leegmaken* die de gegevens en de invoervelden wist, en de Google Map verbergt.
- Zorg ervoor dat de app schaalbaar is. Gebruik het Single responsibility-principe voor componenten. Gebruik frontend services voor communicatie met het backend.

WERKWIJZE:

- **Frontend:** gebruik een CLI van jouw voorkeur om het project te starten. Voeg eventueel aanvullende libraries of packages toe.
- Schets op papier hoe de lay-out eruit komt te zien. Houd rekening met desktopgebruik en mobiel gebruik (bijvoorbeeld via Chrome DevTools).

- Maak de component met twee tekstvakken en een knop. De knop roept een functie aan als `getAddress(postcode, nummer)`, welke de waarden uit de tekstvakken uitleest en doorgeeft aan een service.
- **Backend:** maak een kleine Node.js-server.
 - Deze moet geschikt zijn voor het ontvangen van requests, het eventueel transformeren en doorzetten van de request naar de API's en het terugzenden van de ontvangen data in een response.
 - Gebruik hiervoor bijvoorbeeld de packages `express`, `request`, `node-fetch`, `got`, of nog een andere.
 - Het gebruik van TypeScript is optioneel. Als je dit doet, zorg dan zelf voor goede compilatie.
- Maak een service om de postcode API aan te spreken. Het algemene adres is
 - <http://api.postcodedata.nl/v1/postcode/?postcode=<postcode>&streetnumber=<nummer>&ref=domeinnaam.nl&type=json>
 - Maar er zijn ook andere postcode API's! Zoek er eventueel naar en maak een keuze.
 - Verwerk dynamisch de meegegeven postcode en het huisnummer.
 - Retourneer het resultaat naar de app, die de gevonden data in de userinterface toont.
 - Als geen adres is gevonden, toon dan een melding in de userinterface.
- Vraag een Google Maps API-key aan en gebruik deze in de applicatie. Richtlijnen hiervoor zijn te vinden op <https://developers.google.com/maps/> en op <https://developers.google.com/maps/documentation/javascript/get-api-key?hl=en#key>.
- In het return-resultaat van de hiervoor gebruikte Postcode-API is ook een latitude/longitude-waarde aanwezig. Gebruik deze om een element `<div id="map">` op de pagina te initialiseren. Een voorbeeld hiervan staat bijvoorbeeld (onderin) op <https://developers.google.com/maps/>.
 - Verwerk deze code bijvoorbeeld in een functie `initMap()` die je aanroept nadat het resultaat uit de service is verwerkt.

De app kan er bijvoorbeeld op deze wijze uitzien (hier als webapp):



OPTIONEEL

- Tot slot, optioneel: genereer een distributie-build van je applicatie en publiceer deze op internet. Je kunt hiervoor bijvoorbeeld gebruik maken van gratis hosting op Github Pages, Netlify of een andere Static hosting-partij.
- Stuur de URL naar de docent ter beoordeling ;-)
- In de volgende sessie (28 februari 2022) worden enkele random cursisten gevraagd om zijn/haar project(en) te presenteren.

Veel succes!

Peter Kassenaar, info@kassenaar.com