Uitleg grafieken in angular

Met angular 4 en ngx-charts

Avans Hogeschool Breda

Internettechnologie

Inhoud

[Inleiding 1](#_Toc487188537)

[Voorbereiding en installatie 2](#_Toc487188538)

[Programmeerwerk 3](#_Toc487188539)

[Weerservice 3](#_Toc487188540)

[Weer-‘chart’ 4](#_Toc487188541)

[Aanmaken van de chart 4](#_Toc487188542)

[Koppeling 4](#_Toc487188543)

[Vullen van de chart 4](#_Toc487188544)

[Alternatieve stad opgeven 4](#_Toc487188545)

[Resultaat 5](#_Toc487188546)

[Bibliografie 6](#_Toc487188547)

# Inleiding

Deze handleiding is bedoeld bij de voorbeeldcode van Angular4 in combinatie met de node library ngx-charts.

Er wordt uitgegaan van basiskennis angular, waaronder het maken van een eigen component en service in angular. Hoe dit moet, is te vinden in <http://angular.io>.

# Voorbereiding en installatie

Angular 4 is een framework dat op basis van typescript werkt. Om dit te kunnen programmeren is Node.JS nodig. Dit is te vinden op <https://nodejs.org>. Dit dient als eerste geïnstalleerd te worden.

Voor het gemak wordt Angular CLI gebruikt. Dit is een Command Line Interface voor Angular, zoals <https://cli.angular.io> het beschrijft. Installeren gebeurt door: npm install -g @angular/cli uit te voeren in de command line. Als dit gedaan is, kan het commando ng help uitgevoerd worden, waar aan te zien is wat allemaal kan met Angular CLI. Om snel te starten kun je ng new PROJECTNAAM uitvoeren, waar PROJECTNAAM vervangen dient te worden door de naam van je nieuwe project. Dit is allemaal terug te vinden op <https://github.com/angular/angular-cli>.

Vervolgens moet er in de commandline genavigeerd worden naar de projectmap, cd PROJECTNAAM, en moet npm install uitgevoerd worden om de eerste initialisatie te doen van het project.

Dan moet ngx-charts geïnstalleerd worden: <https://swimlane.gitbooks.io/ngx-charts/content/intro/installing.html>. Dit gebeurt door npm install @swimlane/ngx-charts --save uit te voeren (aangenomen dat de commandline nog steeds in de juiste map zit). Verder is er nog een afhankelijkheid die geïnstalleerd moet worden voor een juiste werking: npm install d3 –save. Nu is de voorbereiding klaar.

Alles in een tabel:

# installatie Angular CLI  
npm install -g @angular/cli  
# nieuw project aanmaken  
ng new PROJECTNAAM  
# initialisatie  
cd PROJECTNAAM  
npm install  
# installatie ngx-charts en afhankelijkheid D3.JS  
npm install @swimlane/ngx-charts --save  
npm install d3 –save

Nu kan er gestart worden met de ontwikkeling. Maak een keuze uit de verschillende IDE’s die typescript ondersteunen. De volgende IDE’s zijn aanbevolen: JetBrains Webstorm en Microsoft Visual Studio Code. De eerste is gratis verkrijgbaar via Avans en de tweede is gratis voor iedereen.

Met het commando ng serve wordt een development webserver gestart die elke keer Typescript compileert zodra er iets opgeslagen wordt. Standaard werkt deze met <http://localhost:4200/>, die te openen is vanaf een willekeurige webbrowser met ontwikkelaarsfunctionaliteiten (F12). Aanbevolen browsers zijn: Google Chrome en Mozilla Firefox.

# Programmeerwerk

Het systeem wat geschreven gaat worden op basis van angular 4 (gegenereerd met de uitleg van het vorige hoofdstuk), bestaat uit twee onderdelen: de service die data binnenhaalt en de chart die de data weergeeft. In dit voorbeeld wordt de actuele weersvoorspelling gebruikt als voorbeelddata.

Ik ben begonnen met het aanmaken van een aantal template bestanden via angular-cli, namelijk de WeatherService en de WeatherChartComponent. Dit gaat als volgt:

# aanmaken WeatherService  
ng generate service Weather

# aanmaken WeatherChartComponent  
ng generate component WeatherChart

Daarna is de service te vinden onder ./weather.service.ts en de component onder ./weatherchart/weatherchart.component.ts

# Weerservice

Er waren drie opties die in aanmerking kwamen voor het aanleveren van het weer:

* Yahoo! Weather
* Openweathermap.org
* Buienradar (RTL Weer)

Van deze drie opties was openweathermap.org het snelst om te vinden en te implementeren, dus is de keuze daarop gevallen. Voor deze dienst moet je een account aanmaken en dan krijg je een API key waarmee je de weersvoorspelling op kunt vragen.

In de documentatie is terug te vinden hoe dit in zijn werk gaat: <http://openweathermap.org/current>

Na het aanmaken van een account en nadat de API key binnen is, kan de service verder gemaakt worden. In de code zijn alle methodes voorzien van commentaar. Dit zijn vrij simpele methodes die verder geen uitleg behoeven.

# Weer-‘chart’

Als de service klaar is, kan gestart worden met de chart zelf. In dit voorbeeld wordt de LineChart gebruikt van ngx-charts.

## Aanmaken van de chart

Het makkelijkst is om te beginnen met het voorbeeld over te nemen van de ngx-charts documentatie op <https://swimlane.gitbooks.io/ngx-charts/content/charts/line-chart.html>. Er is daar een Plunkr voorbeeld aanwezig waarvan de HTML in de template van app.ts overgenomen kan worden in weatherchart.component.html en daarnaast de inhoud van de App class, behalve de constructor. Dit is allemaal terug te vinden in de code onder de weatherchart map.

## Koppeling

Als dat gedaan is, moet er een koppeling gelegd worden tussen de service en de component. Een koppeling kan onder andere gelegd worden door te abboneren op de service met de subscribe methode aan te roepen. Hiervoor moet de service wel als provider in de providers-definiëring, in @Component. Zie commentaar bovenin weatherchart.component.ts.  
requestWeather is de methode die de abbonering doet op de service via de weatherService.doRequest(city).subcribe methode.

## Vullen van de chart

fillChart is de methode die ervoor zorgt dat de chart gevuld wordt. Hierover kan gezegd worden dat de informatie die binnenkomt niet zomaar 1 op 1 doorgeven aan de chart.

Allereest is het handig om te weten dat de chart een array van objecten bevat. Elk object binnen de array stelt een lijn voor. Het object zelf bestaat uit 2 parameters: een ‘name’ en een ‘series’ parameter, wat respectievelijk de naam van de lijn en de array met alle waarden van de lijn is. Verderop wordt de name gevuld met de city name en daarna komt een for-loop waarbij de waarden uit de service gehaald wordt en gepushed wordt richting de series array. De series array verwacht zelf ook weer een object met twee parameters, namelijk weer een ‘name’ en een ‘value’, dat respectievelijk de naam van de waarde en de waarde zelf is.

Omdat het ook mogelijk is een alternatieve stad in te voeren, wordt bovenaan requestWeather de chart uitgeschakeld en aan het eind van de fillChart weer ingeschakeld.

## Alternatieve stad opgeven

Als extra’tje is ook de optie ingebouwd dat een andere stad dan Breda opgegeven kan worden waarna het weer voor die stad te zien is. Dit is terug te vinden in de HTML en de onSubmit methode.

# Resultaat

Het resultaat is een werkende, simpele website waarmee de weersvoorspelling opgevraagd kan worden van elke 3 uur tot 5 dagen vooruit. Dit wordt weergegeven in een grafiek.

# Bibliografie

Google. (2017). Opgehaald van Angular: https://angular.io/

Google. (2017). Opgehaald van Angular CLI: https://cli.angular.io/

OpenWeatherMap, Inc. (2017). Opgehaald van Open Weather Map: http://openweathermap.org

Swimlane. (2017). *ngx-charts - Angular2 and D3js Charting framework*. Opgehaald van https://swimlane.gitbooks.io/ngx-charts/content/