

OEFENINGEN ANGULAR 2 - TECH AVOND ATOS

01 – EIGEN PROJECT

- a) Het algemene Github-adres met het basisproject is github.com/PeterKassenaar/ng2-base
- b) Maak een eigen Hello World-app. Dit kan op verschillende manieren:
 - o Begin met een lege map en voeg zelf `package.json`, `tsconfig.json`, enzovoort toe. De volgorde staat in het bestandje `Angular 2 dependencies.txt`.
 - o OF: gebruik als basis `Angular2-base` en vul zelf aan waar nodig. LET OP: het project dat je download is nog *niet* een werkend project. Je moet op de aangegeven plekken zélf code invoegen.
 - o OF: volg zelf de stappen in de QuickStart-demo op angular.io.
- c) Probeer eens of je een nieuwe component kunt maken en deze bootstrappen. De nieuwe component komt dan *in plaats van* `<hello-world>`, nog niet erbij. Volg deze stappen:
 - o Maak een nieuw bestand, bijvoorbeeld `\app\nieuwe.component.ts`.
 - o Importeer de juiste afhankelijkheden.
 - o Schrijf een selector en een HTML-template.
 - o Pas de bootstrapper aan, zodat daarin nieuwe component wordt geïmporteerd en gestart.
 - o Pas `index.html` aan, zodat de juiste selector wordt gebruikt.
 - o Draai `npm start` om je nieuwe component te testen.

02 – DATABINDING

- a) Breid je app uit oefening 1 (Hello World) uit met een field/property. Bindt deze property in het template dat gebruikt wordt.
 - o Doe dit zowel impliciet (met rechtstreekse initialisatie van variabelen) als expliciet (met aparte declaratie en initialisatie in de `constructor`).
- b) Maak een array van properties. Bind ze in het template met de directive `*ngFor`.
 - o Geef via TypeScript expliciet op dat de property uit een array van strings of een array van objecten moet bestaan (maak zelf deze keuze).
- c) Bind een HTML-knop aan een event en schrijf een event handler.
 - o Gebruik bijvoorbeeld de notatie `<button (click)="myClickHandler()">` in de HTML.
 - o Schrijf in de class de functie `myClickHandler() { ... }`. Toon hiermee een alert in een venster, of (uitgebreider), voeg de inhoud van een tekstvak toe aan de array uit oefening b). Zorg dat deze array weer wordt gebonden in de user interface.

03 - SERVICES

- a) Als het goed is heb je inmiddels een componenten met data. Verplaats deze data vanuit de component naar een service.
 - o Injecteer de service in de hoofdcomponent waarin je de data wilt gebruiken.
 - o Denk aan de property `[providers]` in de `@Component`-annotatie.
- b) Gebruik de alternatieve notatie en importeer je service *app-wide* (in `boot.ts`). Test of alles nog net zo werkt als voorheen.

04 - VERDER GAAN?

- a) Kijk zelf in de documentatie op angular.io/docs/ts/latest/ voor meer mogelijkheden. Nuttig zijn bijvoorbeeld de opties **Developer Guide** en **Cookbook**. Test opties in je eigen app.