

Labs student track

Making a studenttrack survey application

Lab 1. Setting the stage

Vereiste voorinstallatie

- nodejs, te vinden op website -> <https://nodejs.org/en>
- git, te vinden op website -> <https://git-scm.com/downloads>

Stap 1. Clone de git repository naar een lokale plek

Open een command prompt en navigeer naar een willekeurige directory in je omgeving waar je projecten staan, bijvoorbeeld 'c:\projecten'. Clone de git repository naar een lokale bestemming vanaf url <https://github.com/nicojs/angular2-student-track-survey.git>

Het commando voor clonen is 'git clone <url>'.

Als de spullen zijn gekopieerd naar je lokale plek, cd dan in de nieuwe directory:

```
$ cd Angular2_StudentTrack_Survey_RC
```

Voer daarna de volgende opdrachtregel uit:

```
$ git checkout d56d27f
```

Stap 2. Installeer de node_modules en run de startup

Open een NodeJS command prompt en navigeer in de directory 'Angular2_Beta' en in deze directory voor je de volgende commando's uit:

- npm install
- npm start

De typescript compiler draait en zal bij wijzigingen in de bestanden de bestanden opnieuw transpileren naar JavaScript (ES5).

Na deze stappen zal er een melding komen dat de typescript compiler de *.ts files heeft gecompileerd naar JavaScript.

Daarnaast draait er de lite-server webserver draait en zal automatisch de default browser openen en navigeren naar de opgegeven webroot. De indexpagina die daar staat zal worden getoont. Wijzigingen in de code zullen een refresh opleveren in de browser.

Als alles is gelukt, kijk je nu naar de startup opstelling voor onze applicatie.



Lab 2. TypeScript

Vereiste voorinstallatie
voer commando uit:

```
$ git checkout 235cdb5
```

Als je een error krijgt vergelijkbaar met deze:

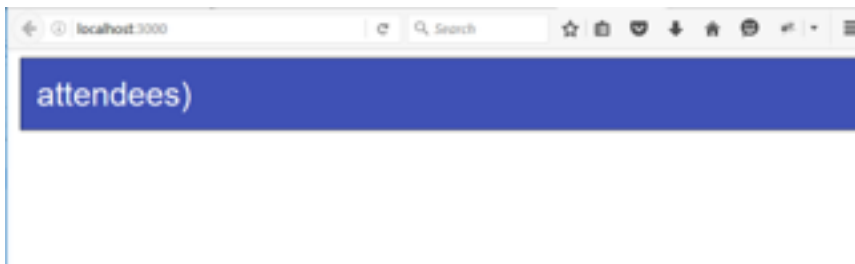
```
error: Your local changes to the following files would be overwritten by checkout:
      src/app/app.js
      src/app/app.js.map
Please, commit your changes or stash them before you can switch branches.
Aborting
```

Voer dan eerst dit commando uit:

```
$ git checkout -- .
```

Voer nu weer de checkout out.

Controleer of het npm start proces nog draait. Er ontbreekt nog wat code dus ziet de applicatie er als het goed is zo uit:



Stap 1. Programmeer de models

Open het bestand `./src/app/models/student.ts` en voeg code toe om de klassen `student` en `studenttrack` te voorzien van gedrag en eigenschappen. Er zijn hints gegeven in de code voor de interface.

De `student` klasse moet worden voorzien van een voornaam, achternaam en een schoolnaam. De `studenttrack` klasse moet worden voorzien van een naam en methoden voor het toevoegen van een student aan de `studenttrack` en een methode voor het ophalen van de studenten van de `studenttrack`.

Stap 2. Programmeer de applicatie met de models

Open het bestand `./src/app/app.ts`. Voorzien de applicatie van een eigenschap `studenttrack`. LET OP: de casing is belangrijk, allemaal kleine letters. De template gebruikt deze naam. Geef de `studenttrack` enkele students en voorzie de `getCount()` methode van een implementatie voor het ophalen van het aantal studenten van de `studenttrack`.

Stap 3. Phat arrows

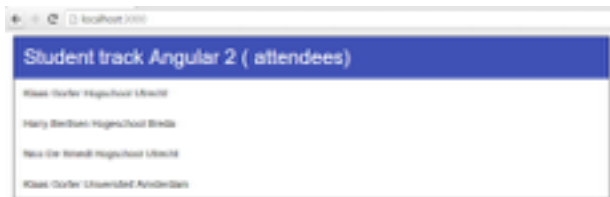
In het geopende bestand (app.ts) staat nu een anonieme functie met een self constructie. Vervang deze constructie met een 'phat arrow' functie waarbij this kan worden gebruikt i.p.v. self.

Sla alle bestanden op en ververs de browser. Als het goed is, wordt er een overzicht getoond met een studenttrack en studenten in de studenttrack. Iedere twee seconden zou een student moeten worden toegevoegd aan de lijst.

Mocht het niet gelukt zijn om de oefening tot een goed einde te brengen, dan kan je een

```
$ git checkout 45e1b6c
```

doen om te kijken naar het eindresultaat.



Lab 3. Components

Vereiste voorinstallatie

Voer commando

```
$ git checkout 8acb355
```

uit in een NodeJS command prompt in de directory 'Angular2_StudentTrack_Survey_RC'. Controleer dat het npm start commando nog draait.

Stap 1. Analyseer de app.ts

Open het bestand ./src/app/app.ts. In dit bestand is nu het divje voor de studentgegevens vervangen door een child component. De manier waarop het component moet worden vormgegeven is te herleiden in het gebruik ervan.

Inventariseer de properties en de events die nodig zijn voor deze childcomponent om deze te kunnen bouwen.

Stap 2. Schrijf de childcomponent

Maak gebruik van de kennis die je nu hebt om een child component te schrijven (app/components/studentdetails.ts), gebruikmakend van de Component decorators op een TypeScript klasse. Als je je werk goed doet, dan werkt na een tijdje de applicatie vanzelf weer.

Mocht het niet gelukt zijn om de oefening tot een goed einde te brengen, dan kan je dit commando doen om te kijken naar het eindresultaat:

```
$ git checkout d4c6a70
```

Student track Angular 2 (6 attendees)	
Hans De Grooth: Angular 2 van A tot Z	
Jelle Jansen: Angular 2 van A tot Z	
Wim Bakker: Angular 2 van A tot Z	
Hans De Grooth: Angular 2 van A tot Z	
Wim Bakker: Angular 2 van A tot Z	
Hans De Grooth: Angular 2 van A tot Z	
Jelle Jansen: Angular 2 van A tot Z	
Wim Bakker: Angular 2 van A tot Z	
Hans De Grooth: Angular 2 van A tot Z	

Lab 4. Pipes

Vereiste voorinstallatie

Voer het volgende commando uit in een NodeJS command prompt in de directory 'angular2-student-track-servey'

```
$ git checkout 211105b
```

Controleer of de het npm start commando nog steeds draait

Stap 1. Schrijf een custom pipe

Bekijk het bestand `./src/app/pipes/studentFormatter.ts`. Schrijf code voor de `studentFormatter` die een student ontvangt en de volgende output genereert voor een student met de getoonde gegevens:

"Naam: John Gorter
School: HAN"

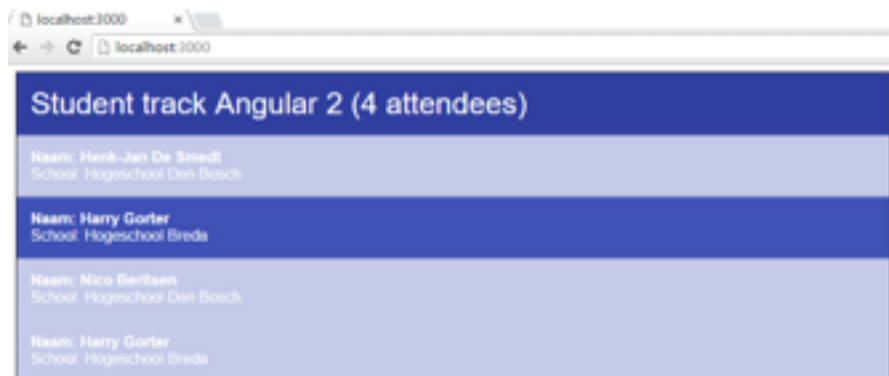
Stap 2. Gebruik de pipe

Open het bestand `./src/app/components/studentdetails.ts` en schrijf code om de eerder geschreven pipe te gebruiken in de applicatie. Merk op dat er HTML uit de pipe komt, wellicht moet je hier even puzzelen om dit goed te tonen.

Als alles goed is, werkt de applicatie naar behoren.

Mocht het niet gelukt zijn om de oefening tot een goed einde te brengen, dan kan je het volgende commando uitvoeren om het eindresultaat te bekijken:

```
$ git checkout ececb45
```



Lab 5. Services

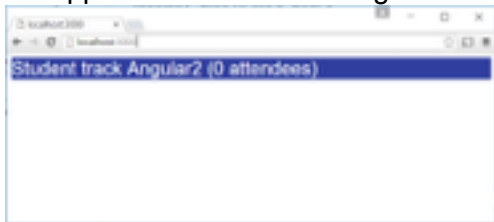
Vereiste voorinstallatie

Voer commando volgende commando uit in een NodeJS command prompt in de directory 'angular2-student-track-survey':

```
$ git checkout e863280
```

Controleer dat het npm start commando draait.

De applicatie moet er als volgt uit zien:



Stap 1. Analyseer de code

Bekijk eerst het bestand `./app/app.ts`. Bekijk hoe deze exact 1 student track op de pagina weergeeft.

Zie ook hoe er een `./api` folder is toegevoegd aan het project. Hierin zitten statische json bestanden die als een mock dienen voor onze backend. Probeer hem eens uit met deze urls: <http://localhost:3000/api/studenttracks/all.json> en <http://localhost:3000/api/studenttracks/1/students.json>

Bekijk daarna de `StudentTrackService` in `./app/services/studenttrackservice.ts`. Zie hoe deze de `studenttracks` ophaalt. Begrijp je hoe het komt dat er nog geen studenten zichtbaar zijn?

Bekijk tot slot bestand `./src/app/app.ts`. Deze werkt momenteel met 1 `StudentTrack`.

Stap 2. Herschrijf de Services

Herschrijf de `StudentTrackService` zodat deze in de constructor een `StudentService` binnenkrijgt. Pas de methode `'getStudentTracks'` aan zodat deze na het ophalen van `StudentTracks` ook de `students` erbij ophaalt. Implementeer de methode `'getStudentsForTrack'` zodat deze de studenten ophaalt die bij de track horen.

Let op dat de `StudentService` nog niet bekend is bij de dependency injection container. Hier zul je dus ook iets voor moeten doen.

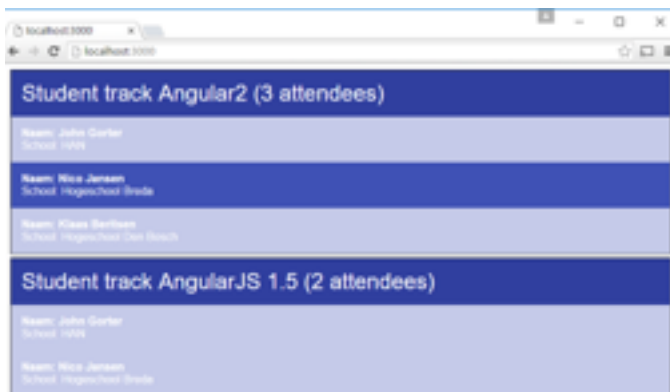
Als je dit goed hebt gedaan zie je op het scherm al studenten bij de `StudentTrack`.

Stap 3. Herschrijf de template

In vervang de `studentTrack` property in `./app/app.ts` in een array van `StudentTrack` objecten, met de naam `studentTracks`. Pas de template dusdanig aan, dat er een `ngFor` gedaan wordt over de `studenttracks`. Let op dat je hier de methode aanroept op de service om alle tracks op te halen.

Mocht het niet gelukt zijn om de oefening tot een goed einde te brengen, dan kan je het volgende commando uitvoeren om het eindresultaat te bekijken:

```
$ git checkout eee5153
```



Lab 6. Forms

Vereiste voorinstallatie

Voer commando volgende commando uit in een NodeJS command prompt in de directory 'angular2-student-track-survey':

```
$ git checkout 84dcc67
```

Controleer dat het npm start commando draait.

De applicatie moet er als volgt uit zien:



Stap 1. Inspecteer de form component

Bekijk het bestand `./src/app/form.ts`. Hier worden twee formulieren opgebouwd. Normaliter zou je dit in twee afzonderlijke componenten doen maar gezien de eenvoud van de applicatie is dit acceptabel. Lees de code die in het component staat en de bijbehorende formulieren goed door.

Momenteel staat er al wel een `<demo-form-sku>` element op de pagina, maar deze directive wordt nog niet gerefereerd in `app.ts`. Open de `app.ts` file en activeer de code die in commentaar staat in deze regel:

```
directives: [StudentDetails/*, StudentTrackForm */]
```

De applicatie werkt nu niet. Maak het formulier af met de juiste template code voor een werkend formulier. De submit van de data werkt nog niet maar dat komt in stap 2.

Stap 2. Implementeer de formulierverwerking

Schrijf de code voor het verwerken van de data. Hiervoor zijn de beide services toegevoegd aan het component. Voeg code toe voor het toevoegen van een studenttrack aan de studenttrack lijst en voeg code toe voor het toevoegen van een geselecteerde student aan een geselecteerde studenttrack.

Normaliter zou je dit ook naar de server sturen om op te slaan. Maar omdat we nu harde json berichten als stub gebruiken voor onze backend kunnen we dit niet doen.

Het toevoegen van student tracks en studenten lijkt foutloos te gaan. Toch zie je ze nog niet op het scherm verschijnen. Wat zou er mis gaan? Ga hiervoor naar stap 3.

Stap 3. Wijzig implementatie van de StudentTrackService

Bekijk de implementatie van de StudentTrackService. Wijzig de implementatie van de service zodat zowel de SurveyApplication als de StudentTrackForm gebruik maken van een gedeelde array voor het bijhouden van de StudentTracks. Wellicht moet je hiervoor ook kleine aanpassingen aan de interface doen van de StudentTrackService.

Mocht het niet gelukt zijn om de oefening tot een goed einde te brengen, dan kan je het volgende commando uitvoeren om het eindresultaat te bekijken:

```
$ git checkout 2ce097f
```



Lab 7. Routing

Vereiste voorinstallatie

Voer commando volgende commando uit in een NodeJS command prompt in de directory 'angular2-student-track-survey':

```
$ git checkout 590115a
```

Controleer dat het npm start commando draait.

De applicatie moet er als volgt uit zien:



Stap 1. Verbeter de app.ts

In deze oefening is een nieuw component opgenomen met de naam StudentCard. Het is de bedoeling dat als er op een student geklikt wordt, dat er dan een StudentCard getoond wordt met de gegevens van de geklikte student. De logica voor het doorgeven en tonen van de juiste student wordt later opgepakt, eerst gaan we de routing naar deze studentcard implementeren.

Bekijk het bestand ./src/app/app.ts. Er is al code toegevoegd voor het importeren van de routing functionaliteit, echter het is nog niet af. Maak de app.ts af met de juiste code voor werkende routing. De volgende zaken zul je moeten regelen:

1. Voorzie de component van de juiste directives voor de template;
2. Verander de template van de SurveyApplication zodat er routerLink en router-outlet attributen en componenten staan voor de verwijzing naar en de presentatie van de studentcard component. Zorg ervoor dat de routerLink ook een studentid heeft opgenomen in de link als een extra segment in de url voor de target.
3. Voeg RouteConfig configuratie data toe aan de component decorator voor matchen en interpreteren van de route. Zorg ervoor dat er een id meegegeven wordt in de url naar de studentcard component.

Hint: In de code is al gemarkeerd waar je je toevoegingen moet doen.

Sla alle wijzigingen op.

We zijn nog maar op de helft van de taken om de routes functioneel te laten zijn. In stap 2 maken we het af.

Stap 2. Verbeter studentcard.ts

De studentcard heeft een student id voor het tonen van deze student in de card. Schrijf code voor het injecteren van de RouteParams object die meegegeven wordt (en geïnjecteerd) in de constructor van de component. Schrijf eveneens de code die gebruik maakt van deze RouteParams en met het student id de call doet naar de studentservice voor het ophalen van de juiste student.

Hint: de code is al gemarkeerd waar je je toevoegingen moet doen.

Sla alle wijzigingen op.

Als alles goed is, werkt de applicatie hierna naar behoren.

Stap 3. Verander van strategie

Controleer of alles naar behoren werkt. Als dit niet het geval is, raadpleeg dan de trainer voor hulp.

Inspecteer met de developer tools de link die wordt gegenereerd op de studentdetails component. Zie hoe er een hashteken gebaseerde link gegenereerd wordt. Sinds HTML5 is er de mogelijkheid om in de history van de browser een stack op te bouwen met urls zonder hash, zonder dat er een page refresh geschied.

We kunnen van deze mooiere urls gebruik maken door PathLocationStrategie te importeren vanuit de 'angular2/router' module en deze in te stellen in de bootstrap als strategie voor de Route_Providers. Verander de bootstrap regel in de onderstaande code:

```
// bootstrap our application..
bootstrap(SurveyApplication, [ROUTER_PROVIDERS,
  provide(LocationStrategy, {useClass: PathLocationStrategy})]);
```

Merk op dat we nu over gaan op de PathLocationStrategy.

Sla alle wijzigingen op en controleer of de applicatie nog werkt. Inspecteer nu de links die worden gegenereerd en merk op dat de hashtekens verdwenen zijn.

Open een detail view van een student en ververs de browser. Kun je dit gedrag verklaren? Wat zou je kunnen doen om dit te voorkomen?

Mocht het niet gelukt zijn om de oefening tot een goed einde te brengen, dan kan je het volgende commando uitvoeren om het eindresultaat te bekijken:

```
$ git checkout 5819611
```

2 studenttrack

studenttrack

Add studenttrack or student to track

Add a new track

Add Student

Add Student

Student track Angular2 (3 attendees)

Name: John Connor
School: MIT

Name: Rick Oshiro
School: Georgetown

Name: Mike Briffon
School: Georgetown

Student track AngularJS 1.5 (2 attendees)

Name: John Connor
School: MIT

Name: Rick Oshiro
School: Georgetown

Go to: 201505

201505: 2015-05-05 10:00 AM - 10:00 AM
Name: 20150505