

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/JohnGorter/StudentTrack_Surveys.git

Het commando voor clonen is git clone 'url'.

Als de spullen zijn gekopieerd naar je lokale plek, voer je de volgende opdrachtregel uit:

```
$ git checkout e7e746
```

Stap 2. Installeer de node_modules en run de startup

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

- npm install
- tsc -p src -w

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.

Open een tweede NodeJS command prompt en navigeer naar dezelfde directory. Voer daar de onderstaande opdracht uit:

- live-server --open="src"

De live-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 `$ git checkout d6054bb` uit in een NodeJS command prompt in de directory 'StudentTrack_Surveys'.

Controleer of de typescript compiler draait en controleer of live-server draait.

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 ipv `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 ae92aee` doen om te kijken naar het eindresultaat.

Lab 3. Components

Vereiste voorinstallatie

Voer commando `$ git checkout 7a1e7c8` uit in een NodeJS command prompt in de directory 'StudentTrack_Surveys'.

Controleer of de typescript compiler draait en controleer of de live-server 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 staat 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, gebruikmakend van de Component annotations op een TypeScript's 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 een `$ git checkout ccaf707` doen om te kijken naar het eindresultaat.

Lab 4. Pipes

Vereiste voorinstallatie

Voer commando `$ git checkout 0d43b76` uit in een NodeJS command prompt in de directory 'StudentTrack_Surveys'.

Controleer of de typescript compiler draait en controleer of de live-server 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:

```
"<b>Naam: John Gorter</b><br>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 een `$ git checkout bd2b12f` doen om te kijken naar het eindresultaat.

Lab 5. Services

Vereiste voorinstallatie

Voer commando `$ git checkout 0506fee` uit in een NodeJS command prompt in de directory 'StudentTrack_Surveys'.

Controleer of de typescript compiler draait en controleer of de live-server draait.

Stap 1. Pas de SurveyApplication aan

Bekijk het bestand `./src/app/app.ts`. Controleer of de applicatie werkt. Inspecteer de toegevoegde ts bestanden in de map 'Services'. Lees deze goed door. In de volgende stappen gaan we deze services gebruiken in de applicatie.

Stap 2. Herschrijf de SurveyApplication

Herschrijf de applicatie zodat deze in de constructor de studenttrackservice krijgt. Open het bestand `./src/app/app.ts` en schrijf code om de services van de studenttrack en students te importeren in de applicatie.

Let op dat de studenttrackservice een `getstudenttracks` methode heeft en we nu meerdere studenttracks hebben om te tonen in de applicatie.

Stap 2. Herschrijf de template

Voeg providers toe aan de component decorator voor de services die je zojuist hebt geïmporteerd en pas de template dusdanig aan, dat er een `ng-for` gedaan wordt over de studenttracks. Let op dat je hier de methode aanroept op de service om alle tracks op te halen.

Als alles goed is, werkt de applicatie naar behoren, maar wacht!! Selecteer nu eens de eerste student? Wat gaat er mis?

Pas de template aan zodat niet alleen de geselecteerde student maar ook de track wordt bijgehouden in de applicatie en zorg dat de selectie weer werkt.

Mocht het niet gelukt zijn om de oefening tot een goed einde te brengen, dan kan je een `$ git checkout e1d46b9` doen om te kijken naar het eindresultaat.

Lab 6. Forms

Vereiste voorinstallatie

Voer commando `$ git checkout a68837f` uit in een NodeJS command prompt in de directory 'StudentTrack_Surveys'.

Controleer of de typescript compiler draait en controleer of de live-server draait.

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.

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.

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

Mocht het niet gelukt zijn om de oefening tot een goed einde te brengen, dan kan je een `$ git checkout 36b56a4` doen om te kijken naar het eindresultaat.