



TypeScript for APIS



Geschreven door Dumoulin Glenn

1. Inhoudsopgave:

Voorblad.....	1
Inhoudsopgave	2
Mijn ervaring van de meetup	3
Voor de meetup.....	3
Na de meetup	3-4
Wat neem ik mee uit de meetup.....	4-5

2. Mijn ervaring van de meetup:

2.1. Voor de meetup:

We hebben de basics van TypeScript wel al eens gezien op het einde van het vak 'Web Programming 5: Mobile Development', maar niet om te zeggen dat we hier veel mee hebben kunnen experimenteren.

Ik weet dus wel dat TypeScript een uitbreiding is van het gekende JavaScript. In TypeScript kan je aan functies meegeven welke type van resultaat je verwacht of van welke type de argumenten zijn. Op deze manier kan je dus fouten vermijden door per ongeluk een getal mee te geven een plaats van een string.

Dit onderwerp heb ik dus ook gekozen omdat ik graag meer wil ontdekken over wat er allemaal te doen is met TypeScript en in dit geval hoe het in zijn werk gaat wanneer we het hebben over APIS.

Enkele dingen die ik mij direct afvraag en waar ik graag meer over hoop te ontdekken tijdens deze meetup zijn:

- Wat de voor- en nadelen zijn van het gebruik van TypeScript voor APIS?
- Wanneer je het best kiest om TypeScript te gebruiken (niet enkel voor APIS)?
- Kan je hier heel uitgebreid mee te werk gaan of is het nog redelijk beperkt en moet het nog uitbreiden in de toekomst? M.a.w. hoever staat de ontwikkeling van deze methodiek?

2.2. Na de meetup:

De spreker in deze meetup was Joshua Goldberg, een frontend developer die werkt bij Codecademy. Het onderwerp van deze meetup was dus zoals ik eerder al vermeld heb TypeScript voor APIS, maar toch viel de meetup een beetje tegen. Het was eerder een infosessie om ons bewust te maken van de verschillende opties die er zijn om TypeScript te gebruiken voor APIS en hoe deze opties onze APIS zouden verbeteren op vlak van stabiliteit, bruikbaarheid en schaalbaarheid. Voor de meetup had ik meer het gevoel dat ik iets compleet nieuw zou gaan ontdekken. Achteraf gezien blijkt toch dat een groot deel mij al bekend in de oren klonk. Begrijp mij niet verkeerd herhaling is altijd goed, ik had mij gewoon voorbereid om meer nieuws te ontdekken vandaar dat de meetup een beetje tegen viel.

De besproken onderdelen werden kort voorgesteld met telkens hun voor- en nadelen volgens Joshua Goldberg, maar er werd nergens dieper op ingegaan dan een korte voorstelling. Het eerste onderdeel, TypeScript, was hierdoor dus niet interessant omdat ik hier al een basis van had gekregen tijdens de opleiding.

Vervolgens ging het over Definition Files en DefinitelyTyped. De uitleg over Definition Files was redelijk onduidelijk waardoor ik na de meetup nog even moest opzoeken wat dit nu eigenlijk was. Dit zijn dus eigenlijk bestanden waarin enkel types gedeclareerd worden en deze types worden dan in de code van de API zelf geïmporteerd en gebruikt. DefinitelyTyped was een vervolg hierop, maar tegelijk ook een tool om het gebruiken van Definition Files te vergemakkelijken. DefinitelyTyped is een van de meest actieve repositories op Github waar de community type definitions voor veelgebruikte libraries zoals bijvoorbeeld React kon publiceren zodat iedereen deze kan gebruiken in hun APIS. Deze types kunnen als aparte npm dependencies geïnstalleerd worden en worden onder de @types folder geplaatst in de node_modules. Indien een package, en ik neem nu weer React als voorbeeld, zelf geen type definitions heeft dan zal TypeScript automatisch in deze @types folder gaan zoeken naar de type definitions voor React. Je installeert dus wel beide packages namelijk gewoon React en @types/react. En in plaats van in de Github repository te gaan zoeken of er een package bestaat waarvoor je er

eentje zoekt, dan kan je de url aka.ms/types even ingeven en hier kan je dan gemakkelijk zoeken of de package die je zoekt bestaat en hoe je deze installeert.

Vervolgens ging de meetup over TypeDoc. Dit is een documentation generator die automatisch jouw TypeScript code omzet naar HTML documentatie. Developers houden ervan als een API goed gedocumenteerd is en ik, als beginnende developer, kan dat zeker bevestigen. TypeDoc is dus een zeer handige tool aangezien het de documentatie update terwijl jij de code aanpast. Op deze manier is je documentatie altijd up-to-date en moet jij als developer niet uit de weg gaan om hier nog eens zelf extra tijd in te steken. Daarbovenop ziet de website er ook nog eens goed uit, rekening houdend met het feit dat deze automatisch gegenereerd wordt, en je kan de documentatie ook zelf customizen door je eigen header en logo toe te voegen bijvoorbeeld. Joshua Goldberg waarschuwt wel dat TypeDoc niet extreem handig is als je weinig tot geen commentaren bij je code plaatst.

De twee volgende onderwerpen, Swagger en HTTP rest, waren minder interessant en nuttig in mijn ogen. Swagger is een tool waarmee de client SDK voor jouw API automatisch kan gegenereerd worden en het ondersteunt zo goed als alle bekende codeertalen en frameworks. Om Swagger te gebruiken beschrijf je jouw API in een JSON bestand en Swagger kan hiermee dan jouw clients genereren. Dit klinkt natuurlijk allemaal heel positief maar de reden waarom ik het niet zo interessant vind, is omdat ik nu na de meetup nog niet begrijp waarom je hiervoor zou kiezen terwijl GraphQL bijvoorbeeld een veel betere optie is. Ook blijkt het zo te zijn dat het handmatig opzetten van Swagger varieert tussen verschillende codeertalen en frameworks. Dit is dan wel iets dat bij meerdere tools voorkomt maar de manier waarop het gezegd werd belooft niet veel goeds. HTTP rest daarentegen is algemeen redelijk gekend maar het is ook allesbehalve een moderne oplossing aangezien je voor elke request een andere API nodig hebt en voor elke actie die je wilt uitvoeren moet je een aparte request voorzien. Dit is waar GraphQL te pas komt en dit was dan ook het laatste onderwerp waar Joshua Goldberg het tijdens deze meetup over had.

GraphQL is voor mij niet onbekend, ik heb hiermee namelijk al projecten gemaakt tijdens deze opleiding. GraphQL is een Query Language die database queries uitvoert in de vorm van requests en de client requests zijn degenen die bepalen wat de in- en output zal zijn. De server zal deze requests dus opvangen en de verwachte output teruggeven. In GraphQL kan je ook schema's aanmaken en deze definiëren de toegestane types voor queries. Deze schema's kunnen heel goed overweg met TypeScript en zijn ook gelijkaardig aan de TypeScript interfaces. Eén van de grootste troeven van GraphQL is dat je slechts één endpoint nodig hebt om al je requests op te vangen. De requests en responses zijn daarbovenop ook nog eens heel flexibel dus je kan zelf bepalen hoe deze in elkaar zitten.

2.3. Wat neem ik mee uit de meetup:

Voor mijn eerste meetup ooit te zijn, heb ik er wel van genoten om eens op een nieuwe manier mijn kennis uit te breiden. De meetup zelf was dan misschien niet wat ik er initieel van verwacht had, maar een enthousiaste gastspreker en enkele nieuwe tools en handige tips hebben er uiteindelijk toch voor gezorgd dat deze meetup voor mij zeker geslaagd was. Ik wil mezelf hier ook niet mee tegenspreken want hoewel de meetup inderdaad wat tegenviel, vind ik nog steeds dat het geslaagd was en dat simpelweg omdat ik mij niet verveeld heb tijdens de meetup en nog steeds heb bijgeleerd. De meetup viel vooral tegen omdat ik op voorhand andere verwachtingen had en niet omdat de meetup slecht of niet interessant genoeg was.

Ik neem dan ook absoluut enkele zaken mee uit deze meetup zoals de TypeDoc tool. Deze lijkt mij heel handig als ik ooit van plan zou zijn om een publieke API te maken die vaak gebruikt zou kunnen worden. Met TypeDoc

kan ik mij dan gewoon focussen op de werking van de API en de documentatie wordt dan automatisch gegenereerd en geüpdatet terwijl ik verder werk of aanpassingen maak.

Wat ik ook interessant vond was het gedeelte over DefinitelyTyped. Dit lijkt mij ook iets nuttig om te blijven onthouden aangezien het een collectie is van type definition files die door de community gemaakt zijn. Je kan deze definition files dus gewoon als npm package installeren in jouw project en je kan er direct mee aan de slag gaan en jouw API verbeteren.

Dit was mijn ervaring van deze meetup. Ik hoop dat degenen die dit gelezen hebben toch ook iets nieuws geleerd of ontdekt hebben en hier misschien iets mee zullen zijn in een toekomstig project.