

Lab 1: Typescript

In onze course bouwen we een contacten applicatie waarbij we gebruik maken van een klasse “Contact” om objecten aan te maken met contact informatie in. Alternatief zou je deze klasse ook kunnen definiëren als interface. In deze lab maken we gebruik van een klasse. De inhoud van deze lab is gebaseerd op een code example van oak3.

Start met het aanmaken van een nieuw project via de AngularCLI:

```
ng new contactApp
```

Open je project in een IDE en maak een folder “models” aan in **/src/app/** en maak hierin een bestand **contact.model.ts** aan. Er zullen 5 properties aangemaakt worden, welke datatypes voorzie je voor elk van deze properties?

Name:

Email:

Phone:

isFavorite:

avatar:

4 van de 5 definiëren we als strings, de isFavorite property krijgt als datatype boolean. Je zou phone kunnen definiëren als datatype number, maar dit geeft problemen als we werken met internationale telefoonnummers.

Als je deze dataset wilt hergebruiken moeten we gebruik maken van een klasse of interface. Maak in het bestand contact.model.ts de klasse Contact aan zoals in het onderstaande voorbeeld:

```
class Contact {  
    name: string;  
    email: string;  
    phone: string;  
    isFavorite: boolean;  
    avatar: string;  
}
```

Vervolgens kan je de klasse gebruiken om objecten te maken, maar properties moeten één voor één gedefinieerd worden:

```
let contact = new Contact();
contact.name = "John";
...
```

Om een object aan te maken met de nodige properties, moeten we de default constructor overschrijven met onze zelf gemaakte constructor. Let op: Je kan default values gebruiken om bepaalde properties als optioneel te markeren. Voeg volgende constructor toe aan je code:

```
class Contact {
    name: string;
    email: string;
    phone: string;
    isFavorite: boolean;
    avatar: string;

    constructor(name: string, email: string, phone: string,
isFavorite = false, avatar = 'assets/avatar.png'){
        this.name = name;
        this.email = email;
        this.phone = phone;
        this.isFavorite = isFavorite;
        this.avatar = avatar;
    }
}
```

Bij isFavorite en avatar voorzien we een default value. Deze zorgt ervoor dat de laatste 2 parameters in de constructor niet verplicht zijn. De default value wordt automatisch gebruikt wanneer deze niet meegegeven worden.

De klasse heeft nu verschillende properties en een constructor, maar kan enkel gebruikt worden binnen zijn eigen typescript file (contact.model.ts). Om ervoor te zorgen dat de klasse ook gebruikt kan worden in bijvoorbeeld onze components, moeten we het export keyword nog toevoegen.

```
export class Contact { ... }
```