IK WIL

Angular Fundamentals Module 3 - Services

vijfhart IT-OPLEIDINGEN

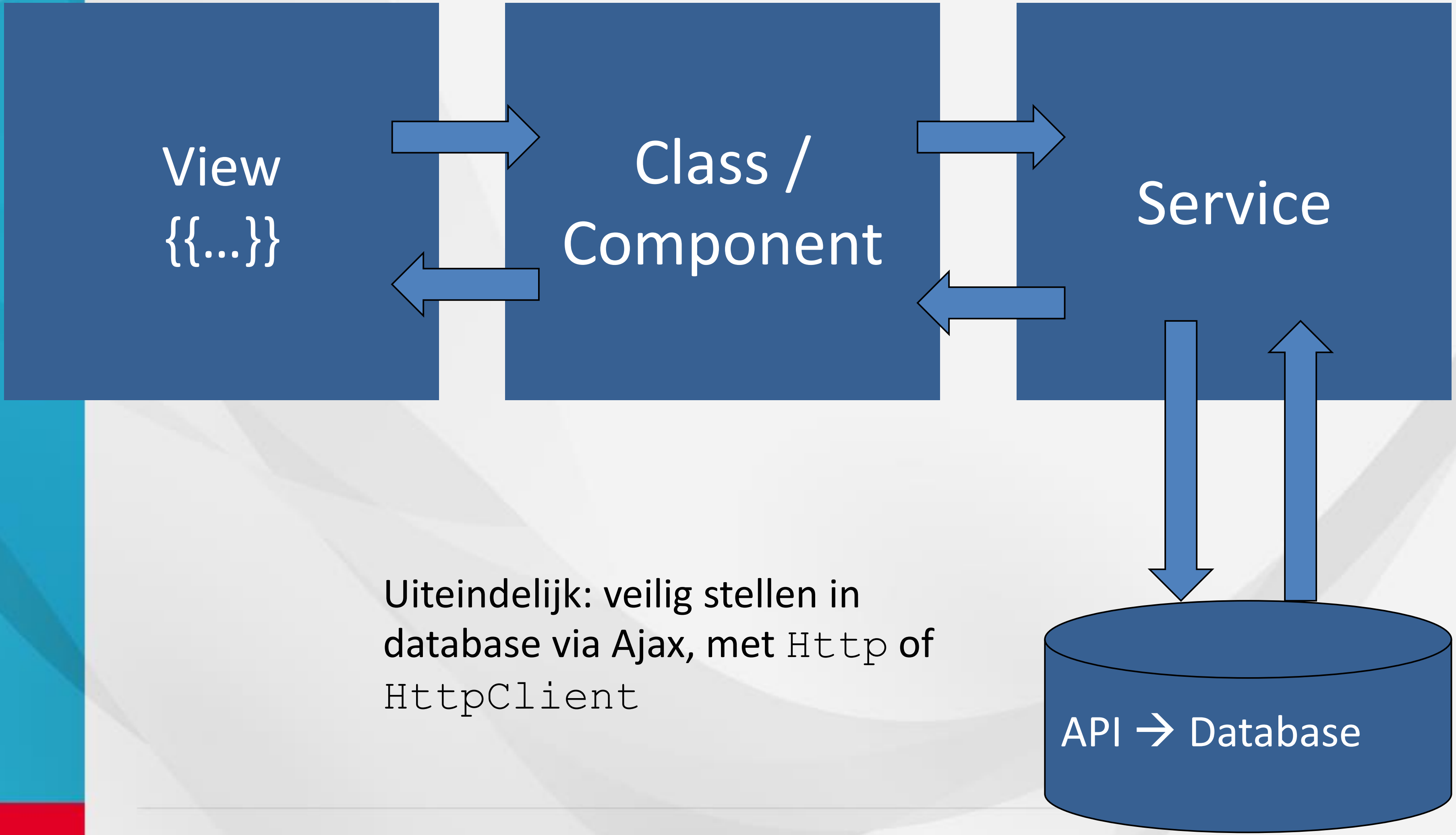
Services

Doel – datafunctionality herbruikbaar maken voor verschillende componenten

- Data retrieval
- Data caching
- Data Storage,
- …

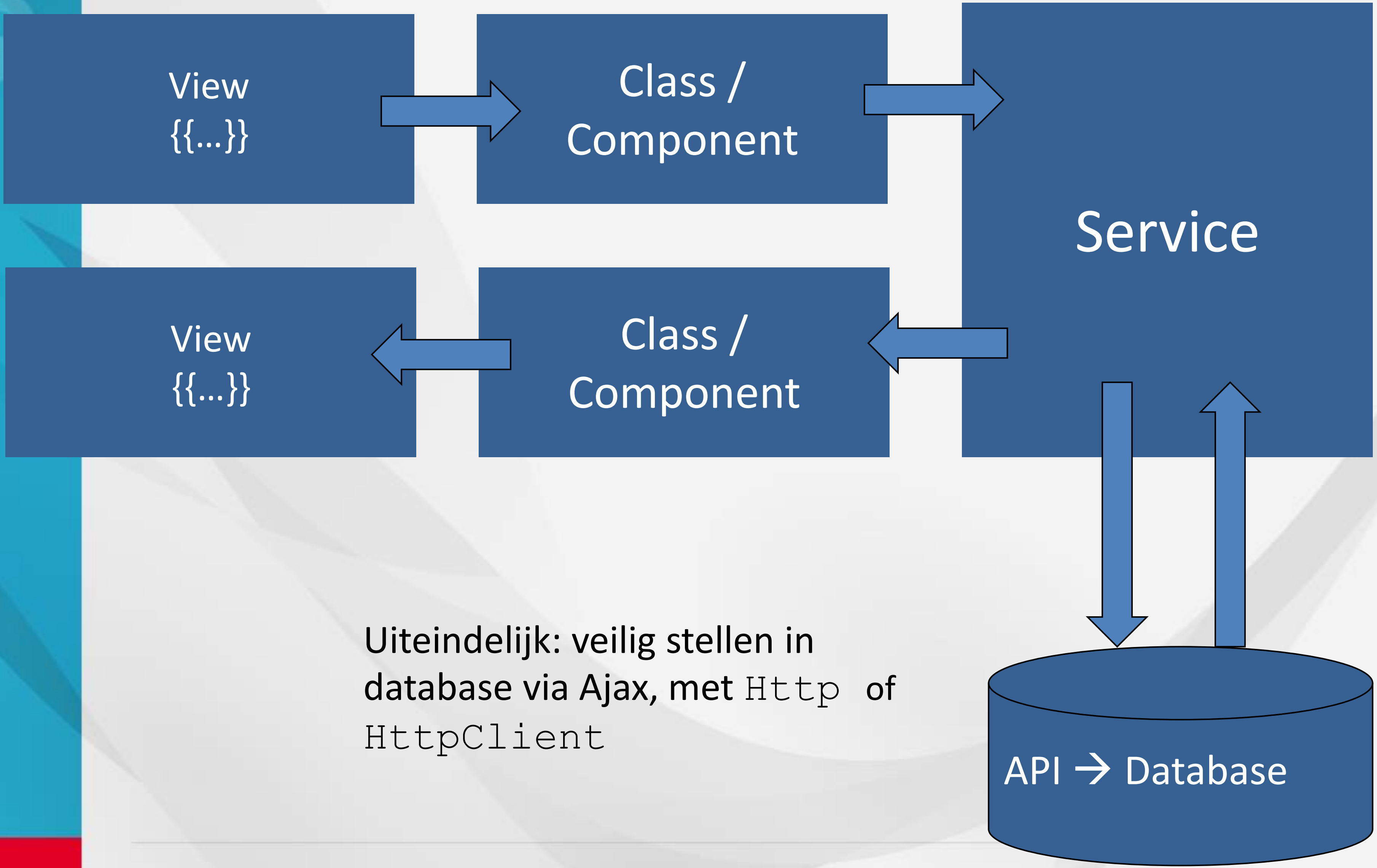
- Angular 2 : één optie
  - `export class myDataService { … }`

Data flow



Uiteindelijk: veilig stellen in database via Ajax, met `Http` of `HttpClient`

Data flow



Uiteindelijk: veilig stellen in database via Ajax, met `Http` of `HttpClient`

Services in Angular 2

Data services in AngularJS:

```
angular.module('myApp')
  .service(…)
  .factory(…)
  .provider(…)
```

Data services in Angular:

```
import {Injectable} from '@angular/core';

@Injectable()
export class CityService{
   //....
}
```

De rol van @Injectable

Why? – Dependency Injection (DI) en metadata!

"TypeScript sees the @Injectable() decorator and emits metadata about our service, metadata that Angular may need to inject other dependencies into this service."

https://angular.io/docs/ts/latest/tutorial/toh-pt4.html

"Our service doesn't have any dependencies at the moment. Add the decorator anyway.

It is a best practice to apply the `@Injectable()` decorator from the start both for consistency and for future-proofing"

Stap 1 – service maken (static data)

```
import { Injectable } from '@angular/core';
import { City } from './city.model'

@Injectable()
export class CityService {
   cities:City[] = [
      new City(1, 'Groningen', 'Groningen'),
      …
   ];

   // retourneer alle cities
   getCities() {
      return this.cities
   }

   // retourneer city op basis van ID
   getCity(id:number) {
        return this.cities.find(c => c.id === id);
   }
}
```

Stap 2 – Service consumeren/injecten

```
…
import {CityService} from "./city.service";

@Component({
   selector   : 'hello-world',
   templateUrl: 'app/app.html',
})

export class AppComponent implements OnInit {
   // Properties voor de component/class
   currentCity: City;
   cities: City[];
   cityPhoto: string;

   constructor(private cityService: CityService) {

   }

   ngOnInit() {
      this.cities = this.cityService.getCities();
   }

   getCity(city: City) {
      this.currentCity = this.cityService.getCity(city.id);
      this.cityPhoto   = `img/${this.currentCity.name}.jpg`;
      console.log('City opgehaald:', this.currentCity);
   }
}
```

Constructor: shorthand voor nieuwe private variable + instantiering!

local variables

Aanroep van de cityService

Instantiation?

- Let op: geen `new()` instantie van de Service!
  - Services zijn Singletons
  - Worden opgehaald uit de Module en/*of* geïnstantieerd in een `constructor()`

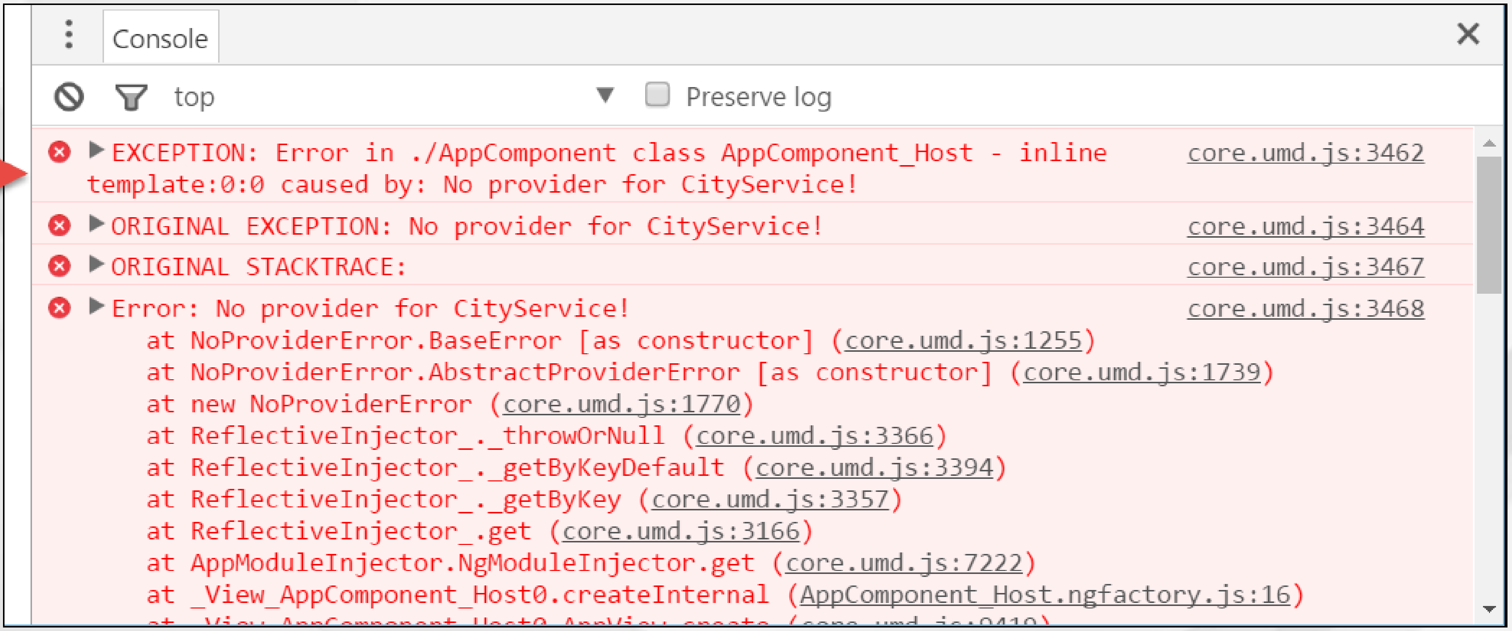
"The constructor itself does nothing.

The parameter simultaneously defines a private cityService property and identifies it as a CityService injection service."

```
constructor(private cityService:CityService) { … }
```

“No provider for CityService”

→ Solution: inject in `app.module.ts`



```
EXCEPTION: Error in ./AppComponent class AppComponent_Host - inline      core.umd.js:3462
template:0:0 caused by: No provider for CityService!
ORIGINAL EXCEPTION: No provider for CityService!                          core.umd.js:3464
ORIGINAL STACKTRACE:                                                      core.umd.js:3467
Error: No provider for CityService!                                       core.umd.js:3468
    at NoProviderError.BaseError [as constructor] (core.umd.js:1255)
    at NoProviderError.AbstractProviderError [as constructor] (core.umd.js:1739)
    at new NoProviderError (core.umd.js:1770)
    at ReflectiveInjector_._throwOrNull (core.umd.js:3366)
    at ReflectiveInjector_._getByKeyDefault (core.umd.js:3394)
    at ReflectiveInjector_._getByKey (core.umd.js:3357)
    at ReflectiveInjector_.get (core.umd.js:3166)
    at AppModuleInjector.NgModuleInjector.get (core.umd.js:7222)
    at _View_AppComponent_Host0.createInternal (AppComponent_Host.ngfactory.js:16)
```

vijfhart IT-OPLEIDINGEN

Service injecteren in Module

→ Alleen de *referentie* naar CityService is niet voldoende.

→ Angular moet de service *injecteren* in de module

→ Gebruik de annotatie `providers: [ … ]`

```
// Module declaration
@NgModule({
    imports     : [BrowserModule],
    declarations: [AppComponent],
    bootstrap   : [AppComponent],
    providers   : [CityService] // DI voor service
})
export class AppModule {
}
```

Array met Service-dependencies

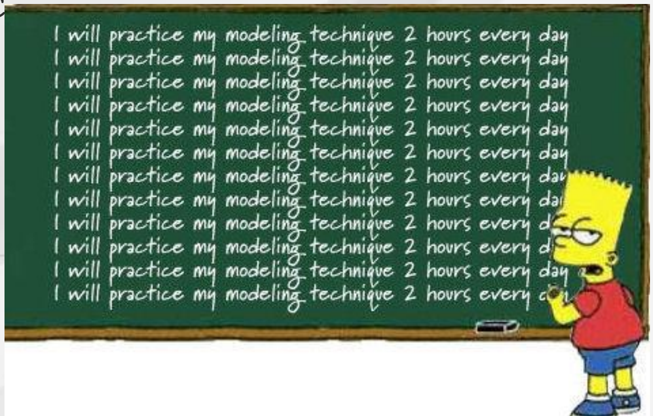
vijfhart IT-OPLEIDINGEN

Singleton?

- Services zijn (in principe) singletons
  - Maar: afhankelijk van de plek waar ze geïnstantieerd worden!
  - Ze zijn een singleton voor de Component/Module en alle child components. Logisch(! / ?)
  - Module/Site-wide gebruiken? (aanbevolen) → Instantieer service in `app.module.ts`

Checkpoint

- Elke service in Angular 2 is een `class`
- Services worden geannoteerd met `@Injectable()`
- Service importeren in de component die hem gebruikt
- Instantiëren of refentie ophalen in `constructor()`
- Service invoegen in de Module bij `providers: []`
- Oefening 5a) + 5b)
- Voorbeeld: `\200-services-static`



Oefening….

vijfhart IT-OPLEIDINGEN