

Web Frameworks

Angular – deel 5

Elektronica – ICT

Sven Mariën

(sven.marien01@ap.be)

Inhoud

1. Deel 1

1. Waarom Angular ?
2. Nieuw Angular project
 - Aanmaken
 - Bestandsstructuur
3. Opbouw
 - Van een Applicatie
 - Van een Component
4. Nieuwe Component Aanmaken

2. Deel 2

1. Interpolation
2. Pipes

3. Deel 3

1. Directives

4. Deel 4

1. Property binding
2. Event binding

5. Deel 5

1. **Two-way binding**
2. Parent-child componenten

Data binding (overzicht)

- **One-way** binding van klasse naar template



- Interpolation: `{{.....}}`
- Property binding: `[.....] = "..."`

Dubbele accolades
Rechte haken

- **One-way** binding van template naar klasse



- Events: `(.....) = "..."`

Ronde haken

- **Two-way** binding (van klasse naar template alsook van template naar klasse)



- Model binding: `[({.....})] = "..."`

"Banana in the box"

5.1. Two-way binding

- 2-way binding wordt gebruikt wanneer de gebruiker input kan geven.
- Een voorbeeld van gebruik is dan ook het “input” veld.

- De syntax voor 2-way binding is als volgt (banana in the box):

```
<input [(ngModel)]="counter"/>
```

- Waarbij er dan in de klasse uiteraard een property wordt voorzien:

```
export class MyComponent {  
  counter: number = 0;  
}
```

- Angular zorgt er voor dat de property en de waarde van het input veld steeds “in sync” blijven.
 - Ofwel wordt de property aangepast wanneer de gebruiker het input veld wijzigt.
 - Ofwel wordt het input veld aangepast indien de property vanuit code wordt aangepast.

Two-way binding (2)

- 2-way binding maakt gebruik van het “**ngModel**” keyword
- Dit is echter gedefinieerd in de **FormsModule**
- Deze dient dus te worden toegevoegd aan de AppModule !

```
<input [(ngModel)]="counter"/>
```

```
imports: [  
  BrowserModule,  
  FormsModule,  
  MDBBootstrapModule.forRoot(),  
  RouterModule.forRoot([  
    { path: "game", component: GameComponent }  
  ]) ]
```

- FormsModule bevindt zich in de **@angular/forms** package

```
import { FormsModule } from '@angular/forms';
```

- Eens deze FormsModule werd toegevoegd, kan “**ngModel**” gebruikt worden in **alle componenten die behoren tot de AppModule**

Two-way binding (3)

- Ander voorbeeld van 2-way binding is een “select” veld.
- Enerzijds kan de gebruiker hier een selectie maken, maar anderzijds willen we ook vanuit de code de selectie kunnen instellen.

Keuze B ▼
Keuze A
Keuze B
Keuze C

```
<select class="form-control" [(ngModel)]="counter">  
  <option value=1>Keuze A</option>  
  <option value=2>Keuze B</option>  
  <option value=3>Keuze C</option>  
</select>
```

```
export class MyComponent {  
  counter: number = 2;
```

- Het Angular framework zorgt er ook hier automatisch voor dat:
 - Als de gebruiker de selectie wijzigt dat de property eveneens wordt aangepast.
 - Als de property vanuit code wordt aangepast, dat de juiste selectie wordt weergegeven.

Inhoud

1. Deel 1

1. Waarom Angular ?
2. Nieuw Angular project
 - Aanmaken
 - Bestandsstructuur
3. Opbouw
 - Van een Applicatie
 - Van een Component
4. Nieuwe Component Aanmaken

2. Deel 2

1. Interpolation
2. Pipes

3. Deel 3

1. Directives

4. Deel 4

1. Property binding
2. Event binding

5. Deel 5

1. Two-way binding
2. **Parent-child componenten**

5.2. “Parent-child” gegevensuitwisseling

- Tussen parent en child componenten is het mogelijk om gegevens uit te wisselen.
- Op een component kunnen we hiervoor de “**extern instelbare**” properties voorzien van een **@Input** decorator. Als parameter kan je (optioneel) een ‘alias’ toekennen aan de property.

```
@Input("theTitle") title: string = "Todo lijstje"
```

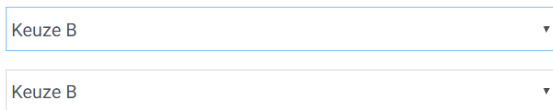
- Deze kan dan vervolgens worden gebonden vanuit zijn ‘parent’ component via property binding (in dit voorbeeld een statische string, maar kan uiteraard evengoed een property van de parent component zijn)

```
<div class="p-col-4">  
  <app-todo-list [theTitle]="'boodschappen'"></app-todo-list>
```

- Ook vanuit de child kunnen gegevens terug worden gestuurd naar de parent (mbv. @Output decorator en een event mechanisme), dit is echter buiten de scope van deze cursus

Oefeningen: Two-way binding

- Ga verder op de Event binding (3) oefening
- Voeg een HTML **select** element toe
- Voeg enkele options toe en bind eveneens aan de '**counter**' property (2-way binding).
- Kopieer de **input** en test.....



- Indien het niet werkt, check even of je de **FormsModule** niet bent vergeten toe te voegen.
- Pas het **input** element aan zodat het ook werkt met 2-way binding.
- En test in combinatie met de dropdowns.....

Two-way binding (2)

- Pas het “welcome” scherm aan zodat er een keuzelijst (select) wordt getoond waarmee de gebruiker rechtstreeks een afbeelding kan selecteren.
- Uiteraard blijven de afbeeldingen nog wel na 5 seconden wisselen en zal de keuzelijst steeds volgen en de huidige getoonde afbeelding weergeven.
- Bouw de keuzelijst op met een **ngFor** aan de hand van een lijst in de klasse



Image 1

Two-way binding (3)

- Stap 1:
 - Voorzie in een component enkele velden om persoonsgegevens in te vullen.
 - Bind deze velden aan een persoons object (werk met een interface) via 2-way binding
 - Geef de ingevulde gegevens onderaan terug weer mbv. interpolation
 - Terwijl de gegevens worden ingevuld, dienen ze dus ook onderaan reeds te verschijnen
- Stap 2:
 - Voorzie een knop 'Bewaren'
 - Geef de ingevulde gegevens pas volledig weer nadat op de knop werd gedrukt

Naam	Voornaam
Straat	Nr
Gemeente	
<input type="button" value="Bewaren"/>	

Bibber	Bert
Veldstraat	102
Zichen-zussen-bolder	
<input type="button" value="Bewaren"/>	

Bert Bibber
Veldstraat 102
Zichen-zussen-bolder

Parent – Child

- Maak 2 componenten:
 - Component 1:
 - Deze component genereert zelf voortdurend willekeurige getallen
 - Het **minimum** en **maximum** getal is instelbaar via publieke properties
 - Het moet ook mogelijk zijn om de component te **stoppen** en te **starten**
 - Component 2:
 - Deze component bevat 2 invulvelden **min:** en **max:**
 - Alsook een **toggle** button (checkbox, switch,..)
 - Component 1 wordt getoond in Component 2
 - de invoer van de gebruiker wordt gebruikt om Component 1 te sturen (min, max & start/stop)

Parent – Child (2)

- Gebruik het Todo lijstje (Angular, deel 4) als Child component.
- Zorg dat je vanuit de parent kan instellen:
 - Title (default: 'Todo lijstje')
 - Max. aantal items in de lijst (standaard = 0, dwz. Onbeperkt)

