Web Frameworks
Angular – deel 4

Elektronica – ICT
Sven Mariën
(sven.marien01@ap.be)

AP HOGESCHOOL ANTWERPEN

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

4.1. Property binding

- Hiermee wordt het mogelijk om in een component eender welke **property** van de HTML objecten in de template te gaan **binden** met gegevens uit de achterliggende klasse. Daarom spreekt met ook wel over ‘data-binding’
- Een voorbeeld.
  - Wanneer we een afbeelding willen weergeven is de bron niet altijd constant, zoals hier wel het geval is:

```
<img src="https://mdbootstrap.com/img/Photos/Horizontal/Nature/4-col/img%20%282%29.jpg" a
```

  - Om de bron te laten koppelen aan een klasse property, kunnen we de “**src**” hieraan “binden”. De syntax voor property binding is als volgt (let op de **rechte haken**):

```
<img [src]="imageUrl"
```

  - Uiteraard dienen we er dan voor te zorgen dat de klasse eveneens deze property heeft:

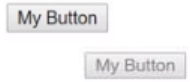
```
export class MyComponent
{
    imageUrl : string;
```

  - Op deze manier is de afbeelding gebonden aan de **imageUrl** property en zal de afbeelding wijzigen als de inhoud van de property wijzigt.

Property binding (2)

- Ander voorbeeld is enable/disable van een “button”:

```
<button [disabled]="buttonStatus">My Button</button>
```



  - De knop zal ‘**disabled**’ zijn indien de property ‘**buttonStatus**’ de waarde TRUE teruggeeft, zo niet is de knop ‘**enabled**’.
  - Aangezien deze property van type ‘boolean’ is, kan je hier **enkel property binding** gebruiken
- Merk op dat:
  - Property binding werkt net zoals Interpolation slechts in 1 richting (= **One-way**).
  - Je kan bovendien **meestal** beide manieren gebruiken:
    - Property binding: `<img [src]="imageUrl"` (rechte haken rond de HTML property)
    - Interpolation: `<img src="{{imageUrl}}"` (dubbele accolades rond de klasse property)
  - Maar let op de juiste syntax:
    - Dit werkt niet met property binding: `<img [src]="https://mdbootstrap.com/img/Photos/Slides/{{imageUrl}}"`
    - Wel met Interpolation: `<img src="https://mdbootstrap.com/img/Photos/Slides/{{imageUrl}}"`
    - Ofwel zo: `<img [src]="'https://mdbootstrap.com/img/Photos/Slides/' + imageUrl"`

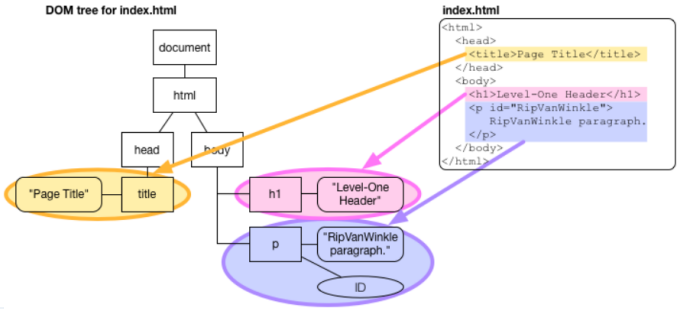
Property binding (3)

- Merk op dat:
  - Property binding in werkelijkheid **niet bind** met de HTML "**attributes**" maar **wel** met de "**properties**" van de overeenkomstige DOM objecten.
  - Soms hebben de "attributes" en "properties" dezelfde naam en merk je het verschil dus niet (bv. **src** bij het image element)
  - Wil je bv. binden met de tekst van een button dan moet dat via de **textContent** Property van het **DOM object.**

textContent

```
<button [textContent]="counter"
```

https://www.w3schools.com/jsref/dom_obj_all.asp



Attributes are defined by HTML. Properties are defined by the DOM (Document Object Model).

- A few HTML attributes have 1:1 mapping to properties. `id` is one example.
- Some HTML attributes don't have corresponding properties. `colspan` is one example.
- Some DOM properties don't have corresponding attributes. `textContent` is one example.
- Many HTML attributes appear to map to properties … but not in the way you might think!

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**

4.2. Event binding

- Mbv. “event binding” kan in een component een event vanuit de HTML template worden opgevangen en afgehandeld door de achterliggende klasse.
- Een voorbeeld:
  - We willen het ‘**click**’ event van een “button” opvangen en er een actie aan koppelen. De syntax voor event binding is als volgt (let op de ronde haken):

```
<button (click)="DoSomething()">Klik mij</button>
```

  - Uiteraard dienen we in de klasse de overeenkomstige functie (method) te voorzien:

```
DoSomething() : void {
  //....
}
```

  - Telkens de gebruiker op de knop klikt zal de functie worden aangeroepen.
  - Welke events kunnen worden opvangen met event binding kan je terugvinden: https://developer.mozilla.org/en-US/docs/web/events

```
<button (mouseover)="DoSomething()"
<label (mouseleave)="DoSomething()">
```

Event binding (2)

- In bepaalde gevallen zijn er extra parameters nodig bij het event:
  - Bv. Positie van de “mousepointer”, “keyboard key”,…
  - Met behulp van ‘$event’ kan event informatie als parameter worden doorgegeven.

```
<button (keydown)="DoSomething($event)"
```

  - Het type van deze parameter is afhankelijk van het event in kwestie:

```
<label (keydown)="DoSomething($event)">
```

```
DoSomething(event: KeyboardEvent): void {
    let keyPressed = event.key
```

```
<label (mousedown)="DoSomething($event)">
```

```
DoSomething(event: MouseEvent): void {
  let xPos = event.clientX
  let yPos = event.clientY;
```

  - Meer info over event types vind je hier: https://www.w3.org/TR/uievents/#mouseevent

Event binding (3)

- Soms is het nuttig om gebruik te maken van “Template reference variables”:
  - Is eigenlijk een “lokale variable” binnen de scope van de HTML template
  - Syntax: (let op het kardinaalteken / hekje / #)

```
<input #in1 type="number" value="5"/>
```

  - Nadien is het mogelijk om bijvoorbeeld de waarde (= value property) van het inputveld elders in de template op te vragen

```
<input #in1 type="number" value="5"/>
<button (click)="DoSomething(in1.value)"
```

  - In de klasse kan de ingevoerde waarde dan verwerkt worden:

```
DoSomething(value : number): void {
```

  - Ook op deze manier kan bv. de gewijzigde waarde van het input veld worden verwerkt:

```
<input #in1 type="number" value="5" (change)="DoSomething(in1.value)" />
<input #in2 type="number" value="5" (input)="DoSomething(in2.value)" />
```

(let op dat in een bepaalde template de variabele naam slechts 1x wordt gebruikt)

Data binding (overzicht)

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

  - Interpolation: {{…..}} Dubbele accolades
  - Property binding: […..] = "…" 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"

Oefeningen: Property binding & Event binding

1. Property binding
   - Welcome component
   - Met willekeurige afbeelding
   - Met wisselende willekeurige afbeelding
2. Event Binding
   - Met button / click event
   - Met template reference value
   - Div element / Mousemove event

Property binding

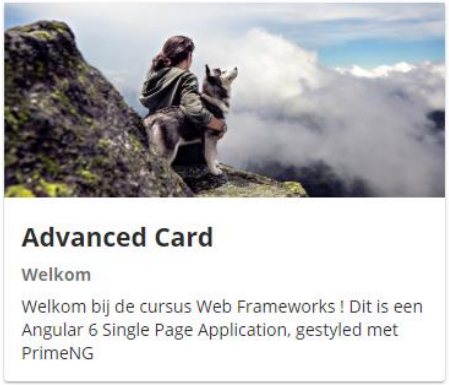
- Maak een “**welcome**” component aan:
  - Deze bevat een “Card” (zie PrimeNG docs, onder de rubriek “Panel”)
  - Neem een ‘Advanced card’, zodat er ook een afbeelding in getoond kan worden.
  - Bind de image **src** aan een klasse property: **imageUrl** (via property binding)
  - Initialiseer de property in de klasse zodat deze een afbeelding weergeeft, bijvoorbeeld:
    - https://mdbootstrap.com/img/Photos/Slides/img%20(1).jpg

Als je PrimeNG nog niet in je applicatie hebt, maak gewoon een component met een afbeelding (img)

De styling kan kan je later nog aanpassen

Property binding (2)

- Pas de “**welcome**” component verder aan:
  - Bind de image **src** aan een klasse property: **imageUrl**
  - Initialiseer de property in de klasse zodat deze steeds een **willekeurige** afbeelding weergeeft, bijvoorbeeld:
    - https://mdbootstrap.com/img/Photos/Slides/img%20(**RND**).jpg
  - Waarbij **RND** een willekeurig getal is tussen 1 en 152



Property binding (3)

- Pas de “**welcome**” component verder aan:
  - Bind de image **src** aan een klasse property: **imageUrl**
  - Initialiseer de property in de klasse zodat deze een **willekeurige** afbeelding weergeeft, bijvoorbeeld:
    - https://mdbootstrap.com/img/Photos/Slides/img%20(**RND**).jpg
  - Waarbij **RND** een willekeurig getal is **tussen 1 en 152**
  - Pas de **imageUrl** property aan zodat deze elke 5 seconden een nieuwe afbeelding weergeeft

Event binding

- Maak de **calculator** component (indien deze nog niet bestond)
- Voorzie de klasse van een:
  - Property: “**counter”** met type **number**, initialiseer op ‘0’
  - 'Method: “**Increment()”,** dewelke de counter met 1 verhoogd
- Voeg een HTML **button** toe aan de template
  - Bind het **click** event aan de **Increment()** functie (event binding)
  - Bind de **textContent** Property aan de **counter** Property van de klasse (prop.binding)
- Test….

- Kopieer ook eens een identieke button bij in de HTML en test het resultaat….
- (Merk op: als je een PrimeNG button maakt, moet je binden met de “**label**” property (en niet met “**textContent**”)

Event binding (2)

- Voeg bij de **button** nog een **input** veld (type=number) toe.
- Bind de **value** hiervan aan de **counter** property (via prop. Binding)
- Test…

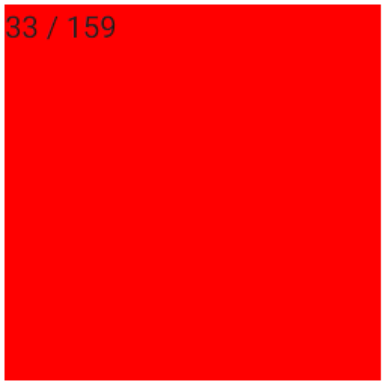
- Wijzig de waarde in het input veld… Is het resultaat zoals verwacht ?

Event binding (3)

- Gebruik het **input** veld en de **button** van de vorige oefening.
- Verwijder echter de bindings
- Zorg er nu voor dat de waarde die de gebruiker heeft ingevoerd wordt doorgegeven naar de klasse **op het moment dat de knop** wordt ingedrukt.
- Gebruik hierbij een “template reference variable” voor het **input** veld.
- Geef de waarde door naar de **counter** property van de klasse mbv. een nieuwe **SetCounter()** method.
- Bind de **textContent** property van de button aan de **counter** property van de klasse.
- Test…

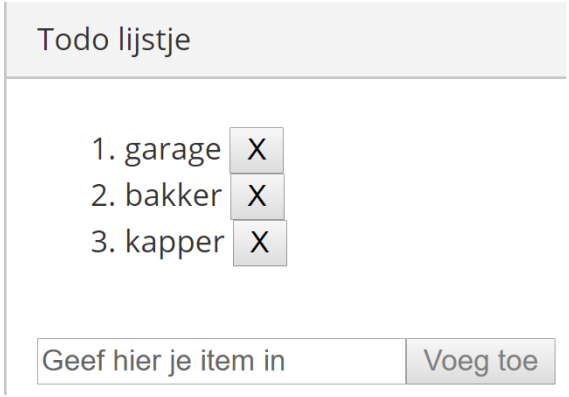
Event binding (4)

- Voeg een HTML **div** element toe
- Style het div element zodat het bv. een breedte & hoogte heeft van 200px; en een background color : Red
- Bind het **mousemove** event met de **DoSomething** functie in de klasse
- In deze functie worden de “mouse coordinates” naar een string geformateerd als: “x / y” en gestockeerd in de property **coords**.
- Toon deze **coords** property in het div element mbv. **Interpolation**
- Test…



Combinatie oef: ToDo lijst

- Maak een component voor een “Todo” lijstje
  - Voorzie een invoer veld en een knop.
  - De knop is uit gegrijsd zolang er niets werd ingegeven in het invoer veld
  - Klik op de knop voegt het nieuwe item toe aan het lijst
  - De lijst wordt weergegeven als een geordende lijst
  - Naast elk item in de lijst verschijnt een knopje om het item terug te verwijderen
  - Indien er nog geen items in je lijst zitten , verschijnt er een tekst in de plaats
    - ‘Je lijst is momenteel leeg’
  - Je zal zeker nodig hebben:
    - ngIf
    - ngFor
    - Property binding
    - Template reference variable
    - Event binding
    - …



Combinatie oef: MoviePlayer (2)

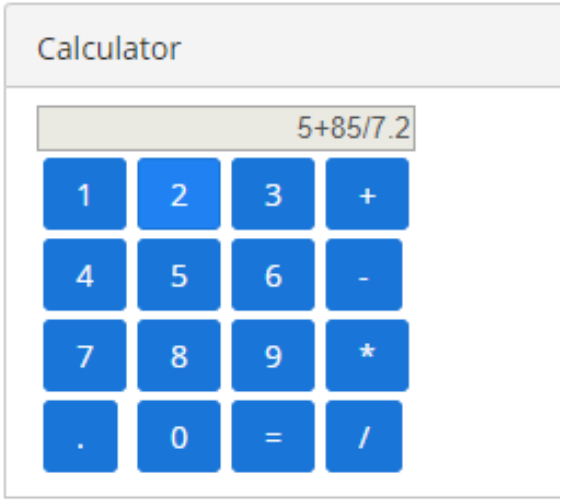
- Maak een nieuwe component
- Maak hierin een media player met een HTML <video> element.
- Vb. film: `<source src="https://file-examples-com.github.io/uploads/2017/04/file_example_MP4_480_1_5MG.mp4" type="video/mp4">`
- Voeg een keuzelijst toe met enkele resoluties (groottes van videoscherm), bv
  - 640x400
  - 800x600
  - …
- De gebruiker kiest een resolutie en de player past zich onmiddellijk aan.

Combinatie oef: MoviePlayer

- Activeer de eigen controls van de player (play knop,.. van de player zelf)
- Voeg daarnaast nog een aparte PLAY en PAUSE knop toe onder de player.
- Wanneer de film bezig is, zorg je dat de eigen PLAY knop niet actief is (uitgegrijsd)
- Wanneer de film in pause staat zorg je dat de eigen PAUSE knop niet actief is.
- Test uiteraard ook of dit ook in orde is als je gebruik maakt van de play/pause knoppen van de player zelf.

Combinatie oef: Calculator

- Maak een calculator component
  - Deze bevat de cijfertoetsen
  - Deze bevat de standaard bewerkingen: +, - , *, /
  - Deze bevat een = toets om het resultaat te bekomen
  - Gebruik de ‘mathjs’ library om de expressie uit te rekenen
  - Genereer de toetsen op een creatieve manier (ngFor,..)



AP HOGESCHOOL
ANTWERPEN