

Dyrektywy w Angular

Tworzenie serwisów Web 2.0

dr inż. Robert Perliński
rperlinski@icis.pcz.pl

Politechnika Częstochowska
Instytut Informatyki Teoretycznej i Stosowanej

28 kwietnia 2019

- 1 Dyrektywy w Angular
 - Dyrektywy atrybutowe
 - Dyrektywa NgClass
 - Dyrektywa NgStyle
 - Dyrektywy strukturalne
 - Dyrektywa NgSwitch
 - Tworzenie własnych dyrektyw
- 2 Źródła

Angular ma trzy rodzaje dyrektyw:

- 1 Komponenty - dyrektywy z szablonem
- 2 Dyrektywy strukturalne - zmieniają strukturę drzewa DOM poprzez dodawanie albo usuwanie elementów z drzewa DOM. Zmieniają strukturę widoku.
- 3 Dyrektywy atrybutowe - zmieniają wygląd albo zachowanie elementu, komponentu albo innej dyrektywy.

Dyrektywy atrybutowe wbudowane w Angular

Dyrektywa atrybutowa zmienia wygląd albo zachowanie elementu drzewa DOM.

Najpopularniejsze dyrektywy atrybutowe wbudowane w Angular:

- 1 NgClass - ustawia klasy elementów drzewa DOM
- 2 NgStyle - ustawia style elementów drzewa DOM
- 3 NgModel - dwukierunkowe wiązanie danych

<https://angular.io/guide/attribute-directives>

Dyrektywa NgClass

Dyrektywa NgClass dodaje lub usuwa klasy CSS do/z elementu HTML.

Klasy CSS są aktualizowane w elemencie w następujący sposób, zależnie od typu wyrażenia przekazanego do dyrektywy:

- `string` - klasy CSS umieszczone w typie `string` (podzielone spacją) zostają dodane
- `Array` - klasy CSS zadeklarowane jako elementy tablicy zostają dodane
- `Object` - kluczami w obiekcie są nazwy klas CSS, które zostają dodane do elementu, jeśli wartość spod danego klucza będzie `true`, w przeciwnym razie (wartość `false`) dana klasa CSS będzie usunięta z elementu HTML

```
<some-element [ngClass]=" 'first second' ">...</some-element>
```

```
<some-element [ngClass]=" ['first', 'second'] ">...</some-element>
```

```
<some-element [ngClass]=" { 'first': true, 'second': false } ">...</some-element>
```

```
<some-element [ngClass]=" stringExp|arrayExp|objExp ">...</some-element>
```

```
<some-element [ngClass]=" { 'class1 class2 class3' : true } ">...</some-element>
```

Dyrektywa NgClass - przykład

Przykład wykorzystania dyrektywy NgClass:

app.component.html

```
...  
<button type="button" (click)="toggleChoice()">NgClass</button>  
...
```

app.component.ts

```
export class AppComponent {  
  ...  
  sentences: Array<string> = [  
    "Ala ma kota", "Nie lubię poniedziałków", "Życie jest super!"];  
  choice: number = 0;  
  
  toggleChoice() : void {  
    this.choice++;  
    if(this.choice>=4) this.choice = 0;  
  }  
}
```

Dyrektywa NgClass - przykład

```
<div>  
  <button type="button" (click)="toggleChoice()">NgClass</button>  
</div>
```

<p>Tutaj poniżej jest zaprezentowana dyrektywa `NgClass`</p>

<p>Wartość zmiennej `choice`, od której zależy wybrany fragment widoku:

```
  <strong>{{choice}}</strong>  
</p>
```

```
<h1 [ngClass]=" 'p-3 mb-2 ' + (choice==0?'bg-primary':'bg-white') ">  
  Nagłówek pierwszego stopnia  
</h1>
```

```
<h2 [ngClass]=" 'p-3 mb-2 ' + (choice==1?'bg-secondary':'bg-white') ">  
  Nagłówek drugiego stopnia  
</h2>
```

```
<h3 [ngClass]=" 'p-3 mb-2 ' + (choice==2?'bg-success':'bg-white') ">  
  Nagłówek trzeciego stopnia  
</h3>
```

```
<h4 [ngClass]=" 'p-3 mb-2 ' + (choice==3?'bg-danger':'bg-white') ">  
  Nagłówek trzeciego stopnia  
</h4>
```

Dyrektywa NgClass - przykład

NgClass

Tutaj poniżej jest zaprezentowana dyrektywa `NgClass`

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: 0

Nagłówek pierwszego stopnia

Nagłówek drugiego stopnia

Nagłówek trzeciego stopnia

Nagłówek trzeciego stopnia

NgClass

Tutaj poniżej jest zaprezentowana dyrektywa `NgClass`

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: 1

Nagłówek pierwszego stopnia

Nagłówek drugiego stopnia

Nagłówek trzeciego stopnia

Nagłówek trzeciego stopnia

NgClass

Tutaj poniżej jest zaprezentowana dyrektywa `NgClass`

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: 2

Nagłówek pierwszego stopnia

Nagłówek drugiego stopnia

Nagłówek trzeciego stopnia

Nagłówek trzeciego stopnia

NgClass

Tutaj poniżej jest zaprezentowana dyrektywa `NgClass`

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: 3

Nagłówek pierwszego stopnia

Nagłówek drugiego stopnia

Nagłówek trzeciego stopnia

Nagłówek trzeciego stopnia

Dyrektywa NgStyle:

- aktualizuje wartość stylów elementu HTML, który ją zawiera
- uswawia jedną lub więcej właściwości stylu określonego jako pary klucz-wartość oddzielone dwukropkiem
- klucz jest nazwą stylu, z opcjonalnym dopiskiem `.<jednostka>` (np. `'top.px'`, `'font.em'`)
- wartość należy wyznaczyć z wyrażenia, które jest przypisane do danego klucza
- wynik wyrażenia, inny niż `null` oznacza wartość danej właściwości stylu w określonych w kluczu jednostkach miary
- wynik wyrażenia równy `null` oznacza usunięcie danego stylu z elementu HTML

Dyrektywa NgStyle

Dyrektywa NgStyle

Ustawienie czcionki pewnego elementu (`some-element`) na wartość wyniku wyrażenia `styleExp`:

```
<some-element [ngStyle]="{'font-style': styleExp}">...</some-element>
```

Ustawienie szerokości pewnego elementu (jednostka to piksele) na wartość zwróconą przez wyrażenie `styleExp`:

```
<some-element [ngStyle]="{'max-width.px': widthExp}">...</some-element>
```

Ustawienie kolekcji stylów pewnego elementu używając wyrażenia (`objExp`), które zwraca pary klucz-wartość:

```
<some-element [ngStyle]="objExp">...</some-element>
```

Dyrektywa NgStyle - przykład

Przykład wykorzystania dyrektywy NgStyle:

app.component.css

```
h2 {  
  color : white;  
  padding: 12pt;  
}
```

app.component.ts

```
export class AppComponent {  
  ...  
  
  colors: Array<string> = ["red", "green", "blue", "purple"];  
  choice: number = 0;  
  
  toggleChoice() : void {  
    this.choice++;  
    if(this.choice>=4) this.choice = 0;  
  }  
}
```

Dyrektywa NgStyle - przykład

app.component.html

```
...  
<button type="button" (click)="toggleChoice()">NgStyle</button>  
...  
<h2 [ngStyle]=  
  "{ 'font-size.%': choice*50+100,  
    'color': 'black',  
    'background-color': colors[choice]  
  }"  
> Treść w nagłówku drugiego stopnia</h2>  
<!-- [style.color]="null" -->
```

NgStyle

Tutaj poniżej jest zaprezentowana dyrektywa NgStyle

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: 0

Treść w nagłówku drugiego stopnia

NgStyle

Tutaj poniżej jest zaprezentowana dyrektywa NgStyle

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: 1

Treść w nagłówku drugiego stopnia

NgStyle

Tutaj poniżej jest zaprezentowana dyrektywa NgStyle

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: 2

Treść w nagłówku drugiego stopnia

NgStyle

Tutaj poniżej jest zaprezentowana dyrektywa NgStyle

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: 3

Treść w nagłówku drugiego stopnia

Dyrektywy strukturalne

Dyrektywy strukturalne:

- są odpowiedzialne za strukturę kodu HTML, za jego układ
- tworzą albo przekształcają strukturę drzewa DOM poprzez dodawanie, usuwanie czy modyfikowanie jego elementów
- stosujemy do konkretnego elementu, który taka dyrektywa przekształca razem ze elementami podrzędnymi, potomkami w drzewie DOM
- łatwo rozpoznać, są poprzedzone gwiazdką (*) jeśli występują jako atrybuty, przykład:

```
<p>Tutaj jest przykład dyrektywy <code>NgFor</code></p>
<ul>
  <li *ngFor="let s of sentences">
    Sentencja: {{s}}
  </li>
</ul>
```

Tutaj jest przykład dyrektywy NgFor

- Sentencja: Ala ma kota
- Sentencja: Nie lubię poniedziałków
- Sentencja: Życie jest super!

<https://angular.io/guide/structural-directives>

Dyrektywy strukturalne wbudowane w Angular

Dyrektywy strukturalne wbudowane w Angular:

- 1 NgIf
- 2 NgForOf
- 3 NgSwitch
- 4 Template Input Variables
- 5 ng-template
- 6 ng-container

Dyrektywa NgSwitch

Dyrektywa NgSwitch

- dodaje albo usuwa szablony kodu HTML (pokazuje albo ukrywa widoki) w zależności do tego, która instrukcja case zostanie dopasowana do wartości przekazanej w wyrażeniu switch
- jest w rzeczywistości zbiorem trzech współpracujących ze sobą dyrektyw: NgSwitch, NgSwitchCase, NgSwitchDefault

```
<container-element [ngSwitch]="switch_expression">
  <some-element *ngSwitchCase="match_expression_1">...</some-element>
  <some-element *ngSwitchCase="match_expression_2">...</some-element>
  ...
  <some-element *ngSwitchDefault>...</some-element>
</container-element>
```

- przykład dla bohaterów:

app.component.html

```
<div [ngSwitch]="hero?.emotion">
  <app-bohater-dobry *ngSwitchCase="'dobry'" [bohater]="bohater"></app-bohater-dobry>
  <app-bohater-zly *ngSwitchCase="'zly'" [bohater]="bohater"></app-bohater-zly>
  <app-bohater-zmienny *ngSwitchCase="'zmienny'" [bohater]="bohater"></app-bohater-zmienny>
  <app-bohater-nieznany *ngSwitchDefault [bohater]="bohater"></app-bohater-nieznany>
</div>
```

Dyrektywa NgSwitch

Dyrektywa NgSwitch

- dyrektywa `[ngSwitch]` umieszczona w jakimś znaczniku będącym kontenerem (np. `div`, `ng-container`) zawiera wyrażenie, do którego trzeba dopasować wartość
- renderowany jest każdy widok, który ma określoną wartość pasującą do wyrażenia
- jeśli nie ma żadnej pasującej wartości, renderowany jest widok spod dyrektywy `ngSwitchDefault`
- elementy w instrukcji `[NgSwitch]`, ale poza jakąkolwiek dyrektywą `ngSwitchCase` lub `ngSwitchDefault` są umieszczane w danym miejscu

Dyrektywa NgSwitch - przykład

Przykład wykorzystania dyrektywy NgSwitch:

app.component.html

```
...  
<button type="button" (click)="toggleChoice()">NgSwitch</button>  
...
```

app.component.ts

```
export class AppComponent {  
  ...  
  sentences: Array<string> = [  
    "Ala ma kota", "Nie lubię poniedziałków", "Życie jest super!"];  
  choice: number = 0;  
  
  toggleChoice() : void {  
    this.choice++;  
    if(this.choice>=4) this.choice = 0;  
  }  
}
```

Dyrektywa NgSwitch - przykład

Przykład wykorzystania dyrektywy NgSwitch:

app.component.html

```
<p>Wartość zmiennej <var>choice</var>, od której zależy wybrany fragment widoku:  
  <strong>{{choice}}</strong>  
</p>  
<article [ngSwitch]="choice">  
  <div *ngSwitchCase="0">  
    Pierwsza sentencja jest uczelniana, o Ali i jej kocie:<br>  
    <textarea rows="3" cols="30">{{sentences[choice]}}</textarea>  
  </div>  
  <div *ngSwitchCase="1">  
    Druga sentencja jest życiowa, taka o realiach:<br>  
    <code>{{sentences[choice]}}</code>  
  </div>  
  <div *ngSwitchCase="2">  
    Trzecia sentencja jest z Ducha Świętego, taka pełna radości:<br>  
    <h3>{{sentences[choice]}}</h3>  
  </div>  
  <div *ngSwitchDefault>  
    Ta sentencja jest wyświetlana, jeśli żadna inna nie została dopasowana:<br>  
    <strong>Lorem ipsum...</strong>  
  </div>  
</article>
```

Dyrektywa NgSwitch - przykład

Przykład wykorzystania dyrektywy NgSwitch

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: **0**

- wartość *choice* równa 0: Pierwsza sentencja jest uczelniana, o Ali i jej kocie:

Ala ma kota

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: **1**

- wartość *choice* równa 1: Druga sentencja jest życiowa, taka o realiach:
Nie lubię poniedziałków

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: **2**

- wartość *choice* równa 2: Trzecia sentencja jest z Ducha Świętego, taka pełna radości:

Życie jest super!

Wartość zmiennej *choice*, od której zależy wybrany fragment widoku: **3**

- wartość *choice* różna od 0, 1, 2: Ta sentencja jest wyświetlana, jeśli żadna inna nie została dopasowana:
Lorem ipsum...

Tworzenie własnej dyrektywy strukturalnej

Tworzenie własnej dyrektywy strukturalnej:

- importujemy dekorator dyrektywy (@Directive)
- importujemy Input, TemplateRef i ViewContainerRef - wszystkie trzy będą potrzebne do każdej dyrektywy
- stosujemy dekorator do klasy, która ma się stać naszą dyrektywą
- określamy atrybut selektora, który identyfikuje dyrektywę stosowaną do danego elementu

```
import {  
  Directive, Input, TemplateRef, ViewContainerRef } from '@angular/core';  
  
@Directive({ selector: '[appAbc]'})  
export class AbcDirective {  
}
```

- wstrzykujemy dwa obiekty do konstruktora dyrektywy:

```
constructor(  
  private templateRef: TemplateRef<any>,  
  private viewContainer: ViewContainerRef) { }
```

Tworzenie własnej dyrektywy strukturalnej

Tworzenie własnej dyrektywy strukturalnej:

- `ViewContainerRef` - referencja na kontener, do którego możemy dodawać nasze komponenty, nasz kod HTML. Utworzony kontener możemy czyścić funkcją `clear()`.

```
this.viewContainer.clear();
```

- `ViewContainerRef` będzie umieszczony w widoku naszej aplikacji w każdym miejscu, w którym jest element drzewa DOM z naszą dyrektywą.
- Metoda `createEmbeddedView()` tworzy nam instancję `EmbeddedView` zawierającą szablon elementu, w którym nasza dyrektywa została dodana.

```
const embeddedView = this.viewContainer.createEmbeddedView(this.templateRef);
```

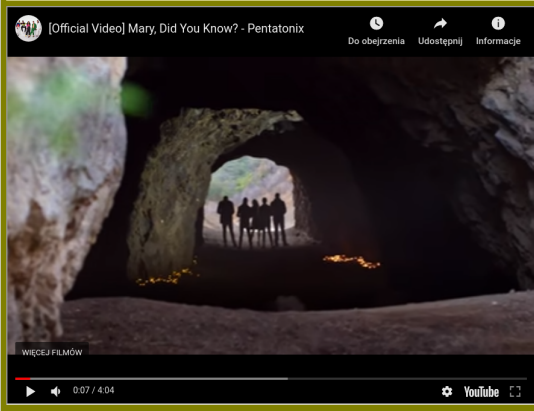
- `TemplateRef` - referencja na szablon kodu HTML. Do konstruktora dyrektywy wstrzykujemy `TemplateRef`. Wykorzystanie naszej dyrektywy w konkretnym elemencie HTML udostępni zawartość (szablon) tego elementu poprzez `TemplateRef`.
- Instancja klasy `ViewContainerRef` ma jeszcze wiele metod: `get()`, `createComponent()`, `insert()`, `move()`, `indexOf()`, `remove()`, `detach()`

Przykład: dyrektywa appYouTube

app.component.html

```
<div *appYouTube="'ifCWN5pJGIE'; width:800; height:600"  
  class="p-2" style="background-color: Olive;">  
</div>
```

Film z portalu YouTube o adresie [ifCWN5pJGIE](https://www.youtube.com/watch?v=ifCWN5pJGIE)



<https://www.youtube.com/watch?v=ifCWN5pJGIE>

Przykład: dyrektywa appYouTube I

Tworzenie własnej dyrektywy appYouTube:

- `ng g d you-tube` albo `ng generate directive you-tube`

```
CREATE src/app/you-tube.directive.spec.ts (229 bytes)
```

```
CREATE src/app/you-tube.directive.ts (143 bytes)
```

```
UPDATE src/app/app.module.ts (1441 bytes)
```

app.module.ts

```
...  
import { YouTubeDirective } from './you-tube.directive';  
  
@NgModule({  
  declarations: [  
    ...,  
    YouTubeDirective  
  ],  
  imports: [  
    BrowserModule  
  ],  
  providers: [],  
  bootstrap: [AppComponent]  
})  
export class AppModule { }
```

Przykład: dyrektywa appYouTube II

Domyślnie utworzony plik dyrektywy `you-tube.directive.ts`:

`you-tube.directive.ts`

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

@Directive({
  selector: '[appYouTube]'
})
export class YouTubeDirective {

  constructor() { }

}
```


Przykład: dyrektywa appYouTube III

Do dyrektyw należy:

- zaimportować odpowiednie dekoratory, klasy, uchwyty do szablonów: `Input`, `OnInit`, `TemplateRef`, `ViewContainerRef`
- dodać odpowiednie pola `@Input`: `appYouTube`, `appYouTubeWidth`, `appYouTubeHeight`
- wstrzyknąć odpowiednie uchwyty do szablu widoku: `templateRef`, `viewContainer`
- dodać kod tworzący znacznik `iframe` dla filmu z YouTube we wstawce programowej `ngOnInit`

Przykład: dyrektywa appYouTube IV

you-tube.directive.ts - po zmianach, działający przykład

```
import { Directive, Input, OnInit, TemplateRef, ViewContainerRef } from '@angular/core';

@Directive({
  selector: '[appYouTube]'
})
export class YouTubeDirective implements OnInit {
  @Input() appYouTube:string;
  @Input() appYouTubeWidth:number = 400;
  @Input() appYouTubeHeight:number = 300;

  constructor( private templateRef: TemplateRef<any>, private viewContainer: ViewContainerRef ) { }

  ngOnInit() {
    this.viewContainer.clear();
    const embeddedView = this.viewContainer.createEmbeddedView(this.templateRef);

    var h2 = document.createElement("h2");
    h2.innerHTML = 'Film z portalu YouTube o adresie
      <a href="https://www.youtube.com/watch?v=${this.appYouTube}">${this.appYouTube}</a>';
    h2.style['background-color'] = 'LemonChiffon'; h2.style['padding'] = '15px';

    var iframe = document.createElement("iframe");
    iframe.width = String(this.appYouTubeWidth);
    iframe.height = String(this.appYouTubeHeight);
    iframe.src = 'https://www.youtube.com/embed/${this.appYouTube}';

    embeddedView.rootNodes[0].appendChild(h2);
    embeddedView.rootNodes[0].appendChild(iframe);
  }
}
```

```
<div
  *appYouTube=" 'o0Vz-ZLaT0U' ; width:400; height:300" >
</div>
```

- <https://angular.io/>
- <https://www.typescriptlang.org/>
- <http://www.angular.love/2017/10/03/angular-dyrektywy-strukturalne/s>