



Dynamic Components Forms Validations

Fravezzi Mattia
m.fravezzi@almaviva.it

Dynamic component

The screenshot shows a web application interface with a header "Hero App" and a list of heroes with "Delete" buttons. Two alert dialogs are overlaid on the interface, connected by red arrows pointing to a central white box.

Alert 1 (Left): Non hai i permessi di amministrazione!

Alert 2 (Right): Sei sicuro di voler eliminare questo eroe?
 Annulla Conferma

Hero List:

- Ant-Man Delete
- Capitan America Delete
- Vedova Nera Delete
- Hulk Delete
- Iron Man Delete
- Thor Delete

Text: alert works!

Diagram: Two red curved arrows originate from the "Delete" buttons of "Ant-Man" and "Iron Man". They point towards a central white rectangular box, indicating a dynamic component that can be added or removed based on user actions.

Dynamic component

```
<div class="ad-banner-example">
  <h3>Advertisements</h3>
  <ng-template adHost></ng-template>
</div>
```

```
@Directive({ selector: '[adHost]' })
export class AdDirective {
  constructor(
    public vcr: ViewContainerRef
  ) { }
```

```
@Component()
export class AdBannerComponent implements {
  @Input() ads: AdItem[] = [];

  @ViewChild(AdDirective, { static: true }) adHost!: AdDirective;

  constructor( private cfr: ComponentFactoryResolver) { }

  loadComponent() {
    const componentFactory =
      this.cfr.resolveComponentFactory(adItem.component);
    const vcf = this.adHost.viewContainerRef;
    vcf.clear();
    const ref = vcf.createComponent<AdComponent>(componentFactory);
    ref.instance.data = adItem.data;
  }
}
```

Dynamic component

```
@Injectable()
export class AdService {
  getAds() {
    return [
      new AdItem(HeroProfileComponent, { name: 'Bombasto', bio: 'Brave as they come' }),

      new AdItem(HeroJobAdComponent, {
        headline: 'Openings in all departments',
        body: 'Apply today'
      }),
    ];
  }
}
```

DOM Decorator

- `@ViewChild()`
- `@ViewChildren()`
- `@ContentChild()`
- `@ContentChildren()`
- `QueryList<Item>`

Custom Pipes

```
@Pipe({
  name: 'truncate'
})
export class TruncatePipe implements PipeTransform {
  transform(value: any, length: number, symbol: string) {
    return value.split(' ').slice(0, length).join(' ') + symbol;
  }
}
```

```
{{ item.name | truncate:4:'...' }}
```

Template Variables

- a DOM element within a template
- a directive
- an element
- TemplateRef
- a web component

Syntax

```
<input #phone placeholder="phone number" />
<button (click)="callPhone(phone.value)">Call</button>

<form #f>
  <input class="form-control" name="name" ngModel required />
  <button type="submit">Submit</button>
</form>
```


Reactive Forms vs Template Driven Forms

- Flessibilità
- Realizzazione form Complessi
- Unit Test su logica di gestione dei form
- Costruzione Dinamica di Forms
- Rilevamento di modifiche attraverso observable
- Forms Semplici senza elaborazioni
- Accesso diretto sui dati

Template Driven Forms

```
<form #f="ngForm" (submit)="add(f)" [ngClass]="{'error': f.invalid && f.dirty}">
  <input class="form-control" name="name" #nameRef="ngModel" placeholder="Name" [ngModel]="hero?.name"
required minlength="3" [ngClass]="{'error': nameRef.invalid && nameRef.dirty}" />

  <div *ngIf="nameRef.invalid && (nameRef.dirty || nameRef.touched)" class="alert alert-danger">
    <small *ngIf="nameRef.errors?.required">
      Name is required.
    </small>
  </div>

  <button type="submit" [disabled]="f.invalid">Submit</button>

</form>
```

Reactive Forms

```
<form [formGroup]="myForm" (ngSubmit)="onSubmit()">
  <label>
    Name:
    <input type="text" formControlName="name">
  </label>
  <div *ngIf="nameForm.invalid && (nameForm.dirty || nameForm.touched)" class="alert alert-danger">
    <div *ngIf="nameForm?.errors?.required">
      Name is required.
    </div>

  </div>
  <button type="submit" [disabled]="myForm.invalid">Submit</button>
</form>
```

Reactive Forms

```
this.myForm = this.fb.group({  
  name: ['', [Validators.required, Validators.minLength(3), forbiddenNameValidator(/regex/i)]],  
  ability: ['', Validators.required],  
}, { validators: formCrossValidator });
```

DEMO



almaviva.it