FEBRUARY 2024

@THECODEMITTER

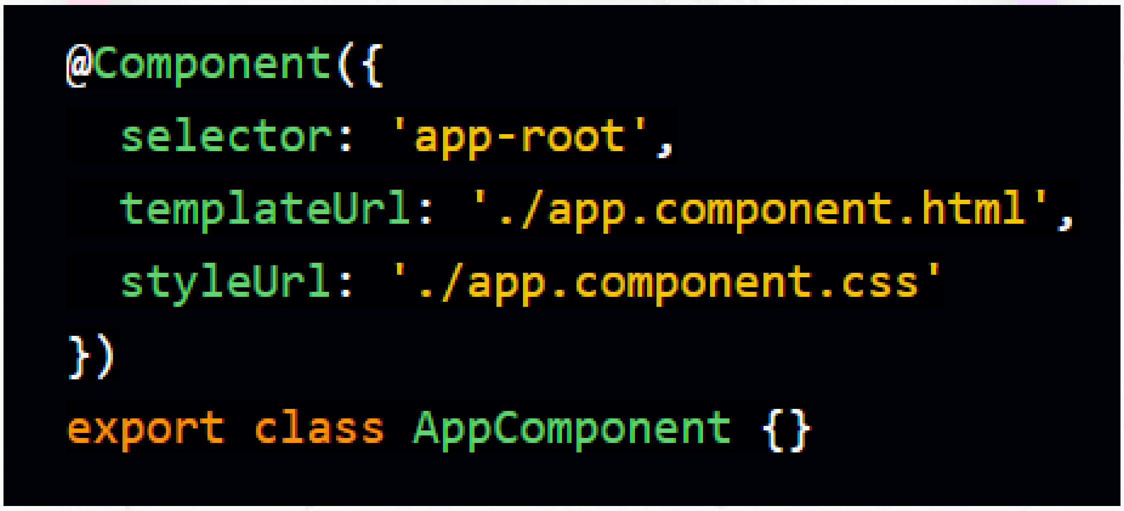
ANGULAR 17 FEATURES

PRESENTED BY
Souvik Mitra

SWIPE →

@Component annotation

A new property **'styleUrl'** has been introduced and it takes a string value. Previously, we only had **'styleUrls'** property which takes an array of strings.



```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {}
```

Application builder

On previous versions, we have the below 2 builds:

- **@angular-devkit/build-angular:browser** to build a client-side rendering application
- **@nguniversal/builders:ssr-dev-server** to build a server-side rendering application.

In Angular 17, we have a new builder which can be used to build both a client-side and a server-side applications:

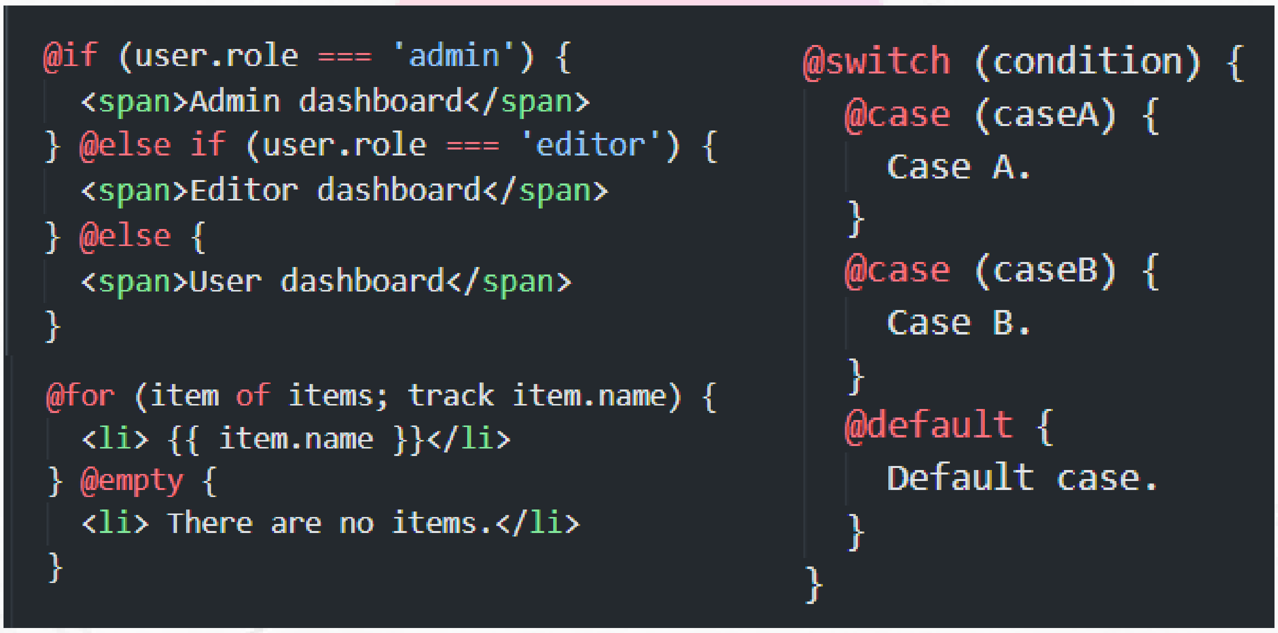
@angular-devkit/build-angular:application

Angular Control Flow

In previous versions, we used ngIf, ngSwitch, ngFor in the template to manage the structure.

Now, here we have a different approach

- **@if** replaces ***ngIf**
- **@for** replaces ***ngFor**
- **@switch** replaces ***ngSwitch**



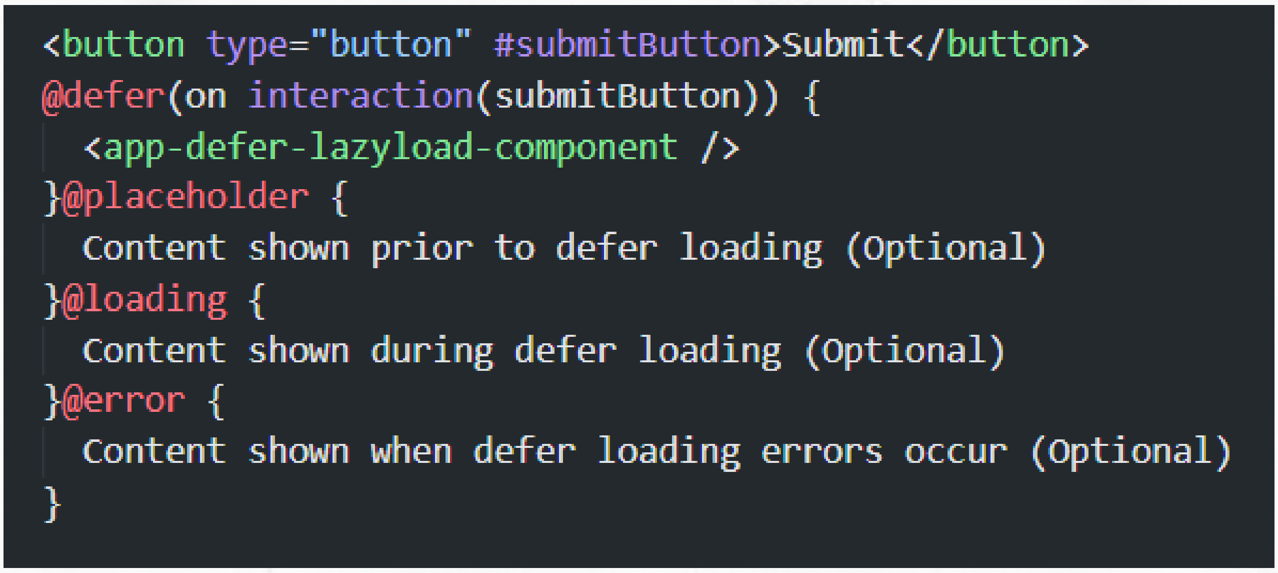
```
@if (user.role === 'admin') {
  <span>Admin dashboard</span>
} @else if (user.role === 'editor') {
  <span>Editor dashboard</span>
} @else {
  <span>User dashboard</span>
}

@for (item of items; track item.name) {
  <li> {{ item.name }}</li>
} @empty {
  <li> There are no items.</li>
}
```

```
@switch (condition) {
  @case (caseA) {
    Case A.
  }
  @case (caseB) {
    Case B.
  }
  @default {
    Default case.
  }
}
```

Defer

Angular 17 has brought up a new way to **lazy load** components, directives and pipes used inside a component template which is **'@defer'**.

With the release of **@defer, standalone components** are now better solution for every possible use-case.



```
<button type="button" #submitButton>Submit</button>
@defer(on interaction(submitButton)) {
  <app-defer-lazyload-component />
}@placeholder {
  Content shown prior to defer loading (Optional)
}@loading {
  Content shown during defer loading (Optional)
}@error {
  Content shown when defer loading errors occur (Optional)
}
```

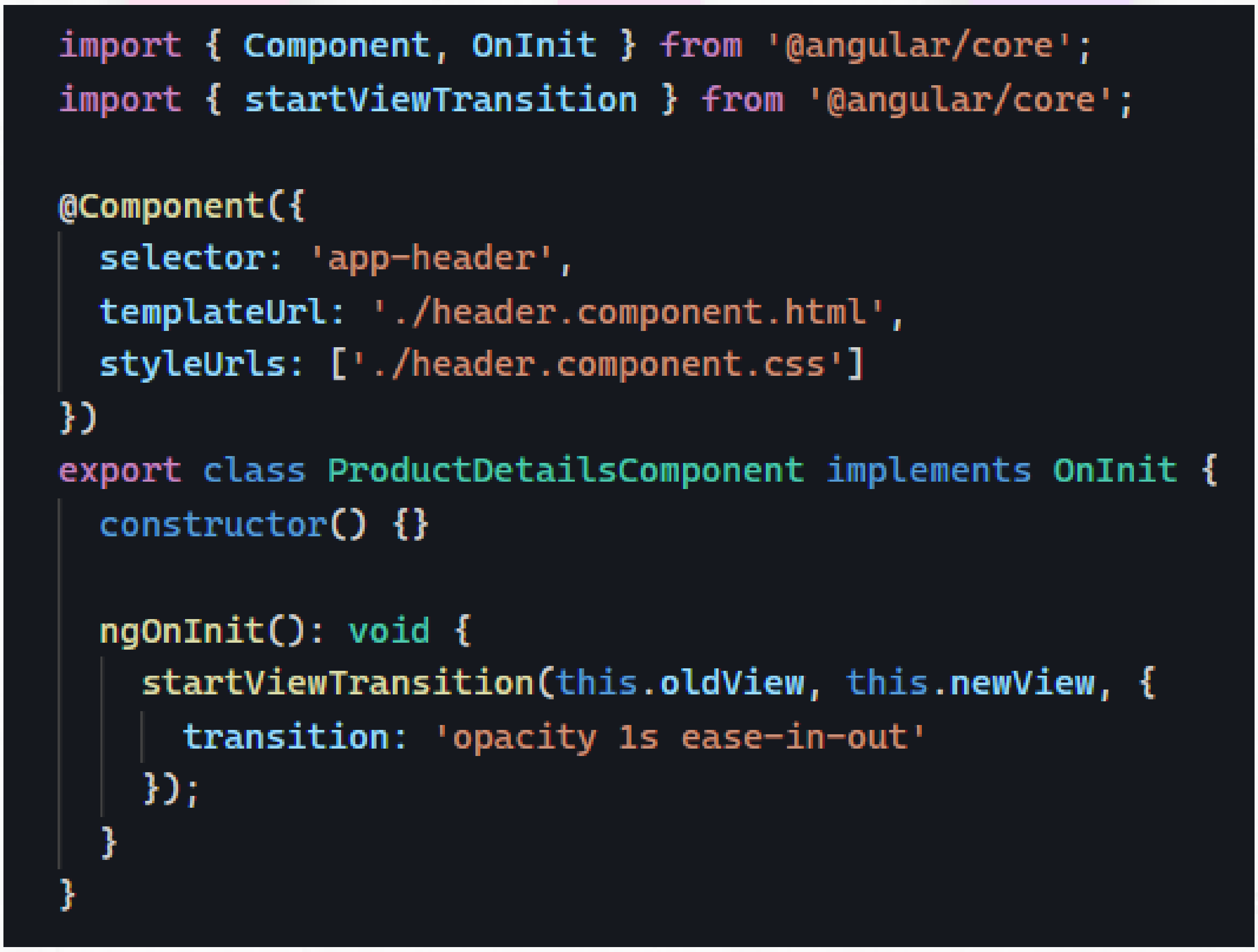
Trigger

Trigger provides conditions when to trigger a **defer loaded** components.

- **on idle:** when the browser reports idle state (default)
- **on viewport(<elementRef>?):** when the element enters the viewport
- **on interaction(<elementRef>?):** when clicked, touched, or focused
- **on hover(<elementRef>?):** when element has been hovered
- **on immediate:** when the page finishes rendering
- **on timer(<duration>):** after a specific timeout
- **when <condition>:** on a custom condition

View Transitions API

Angular 17 has introduced **View Transitions API** that helps in better seamless **animations** on **transitioning** between the pages.

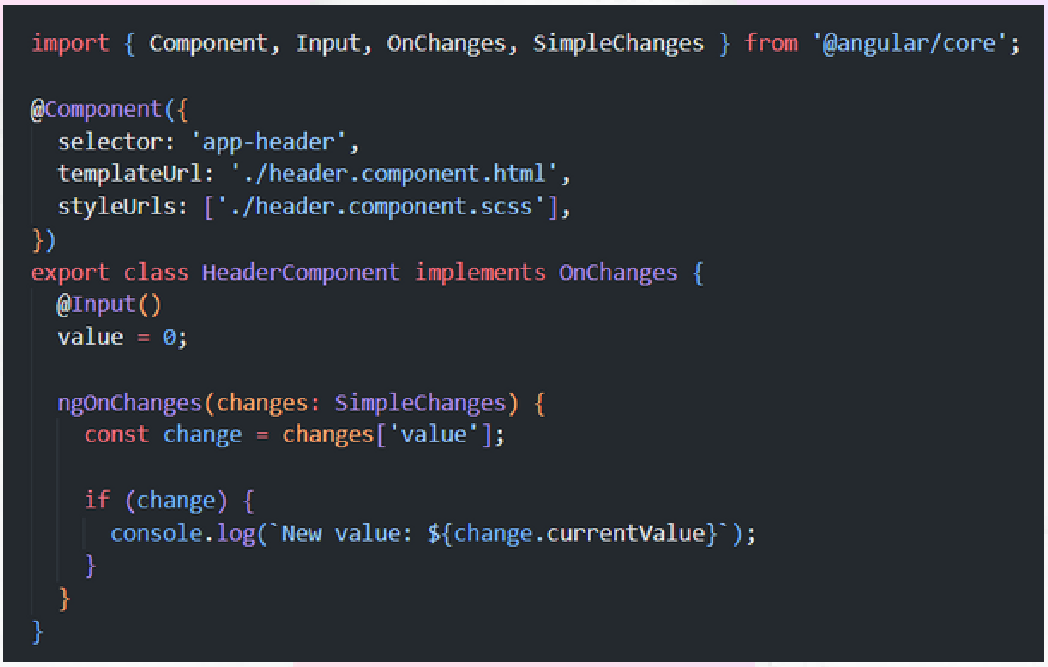


```
import { Component, OnInit } from '@angular/core';
import { startViewTransition } from '@angular/core';

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.css']
})
export class ProductDetailsComponent implements OnInit {
  constructor() {}

  ngOnInit(): void {
    startViewTransition(this.oldView, this.newView, {
      transition: 'opacity 1s ease-in-out'
    });
  }
}
```

Input as Signal

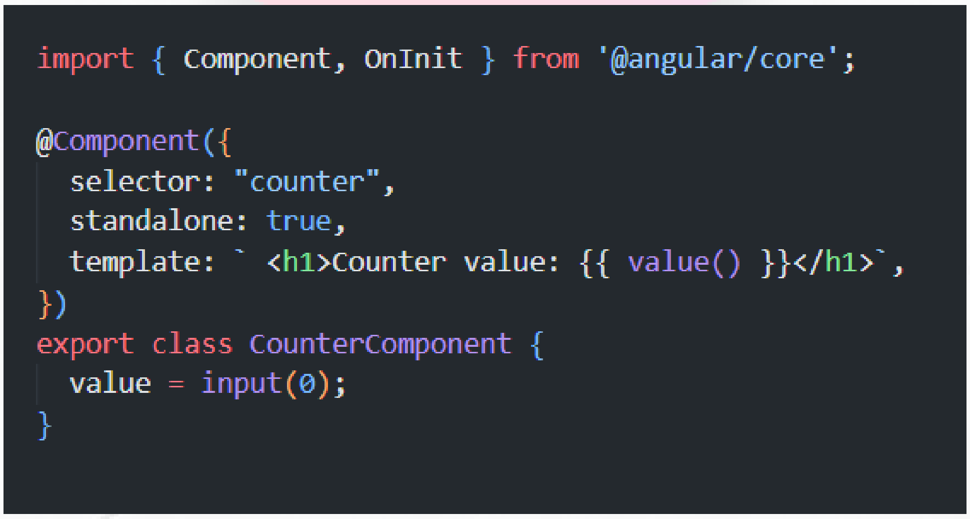
In **Angular 17**, converting **@Input** to a **Signal Input** is the new feature.

Initially, we used **@input** like below:



```
import { Component, Input, OnChanges, SimpleChanges } from '@angular/core';

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.scss'],
})
export class HeaderComponent implements OnChanges {
  @Input()
  value = 0;

  ngOnChanges(changes: SimpleChanges) {
    const change = changes['value'];

    if (change) {
      console.log(`New value: ${change.currentValue}`);
    }
  }
}
```

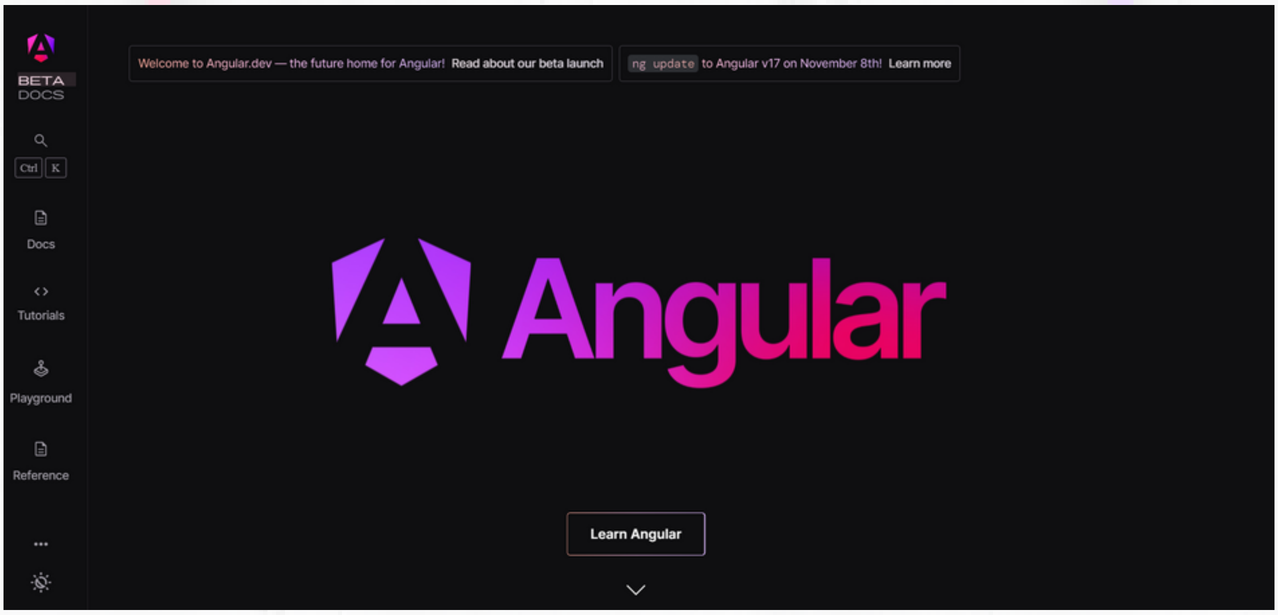
Now, we use **input** as below:



```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: "counter",
  standalone: true,
  template: ` <h1>Counter value: {{ value() }}</h1>`,
})
export class CounterComponent {
  value = input(0);
}
```

New look

A fresh new UI has come up with Angular 17. Have a look in the new website. Just open and start scrolling, you'll get amazed. Along with this new trendy UI, it has brought in a modern version of Angular. Let's learn Angular in a new way.


BETA
DOCS
Ctrl K
Docs
Tutorials
Playground
Reference
Welcome to Angular.dev — the future home for Angular! Read about our beta launch
ng update to Angular v17 on November 8th! Learn more
Angular
Learn Angular

SM

SOUVIK MITRA

thecodemitter.bio.link