



Save it



Like it



v18

# Deferrable Views for Lazyloading



@coder\_aishya



## Deferrable views

delays the loading or initialization of a component or module until specific conditions are met

## Prerequisites for using Deferrable Views

- Angular 17 :

Application should be of Version 17 or later because deferrable views are introduced in Angular version 17

- Standalone Components:

only standalone components, directives, and pipes can be deferred.



@coder\_aishya



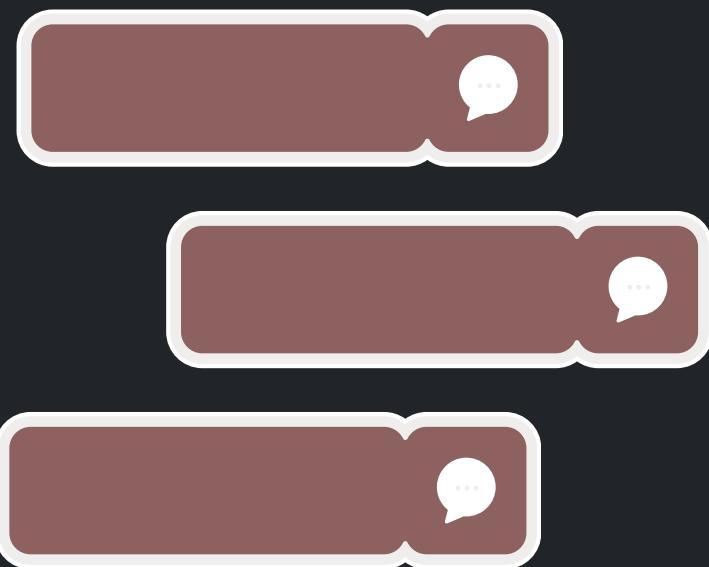
## How they work?

- Wrap a section of your template in a @defer block:  
This tells Angular that the contents of this block should not be loaded immediately.
- Specify triggers for when the deferred content should be loaded. These triggers can be based on user interaction, scroll position, timeouts, or custom conditions.
- While the deferred content is waiting to be loaded, you can display a placeholder. This could be a simple text message, a spinner, or any other kind of content you like.
- Once the trigger conditions are met, the deferred content is loaded and rendered.



@coder\_aishya





## LazyLoad Comments Section



```
@defer {  
  <comment-list />  
}
```

Wrap the comments  
component inside @defer  
decorator



@coder\_aishya



## @defer blocks for loading and error states

```
● ● ●  
@defer {  
  <comment-list/>           content to be shown while loading  
} @loading {  
    
} @error {  
  <p> Loading failed :( </p>    error content  
} @placeholder {  
    
}  
                                placeholder content
```



@loading block : declare content that will be shown during the loading of any deferred dependencies



@placeholder block : optional block that declares content to show before the defer block is triggered.



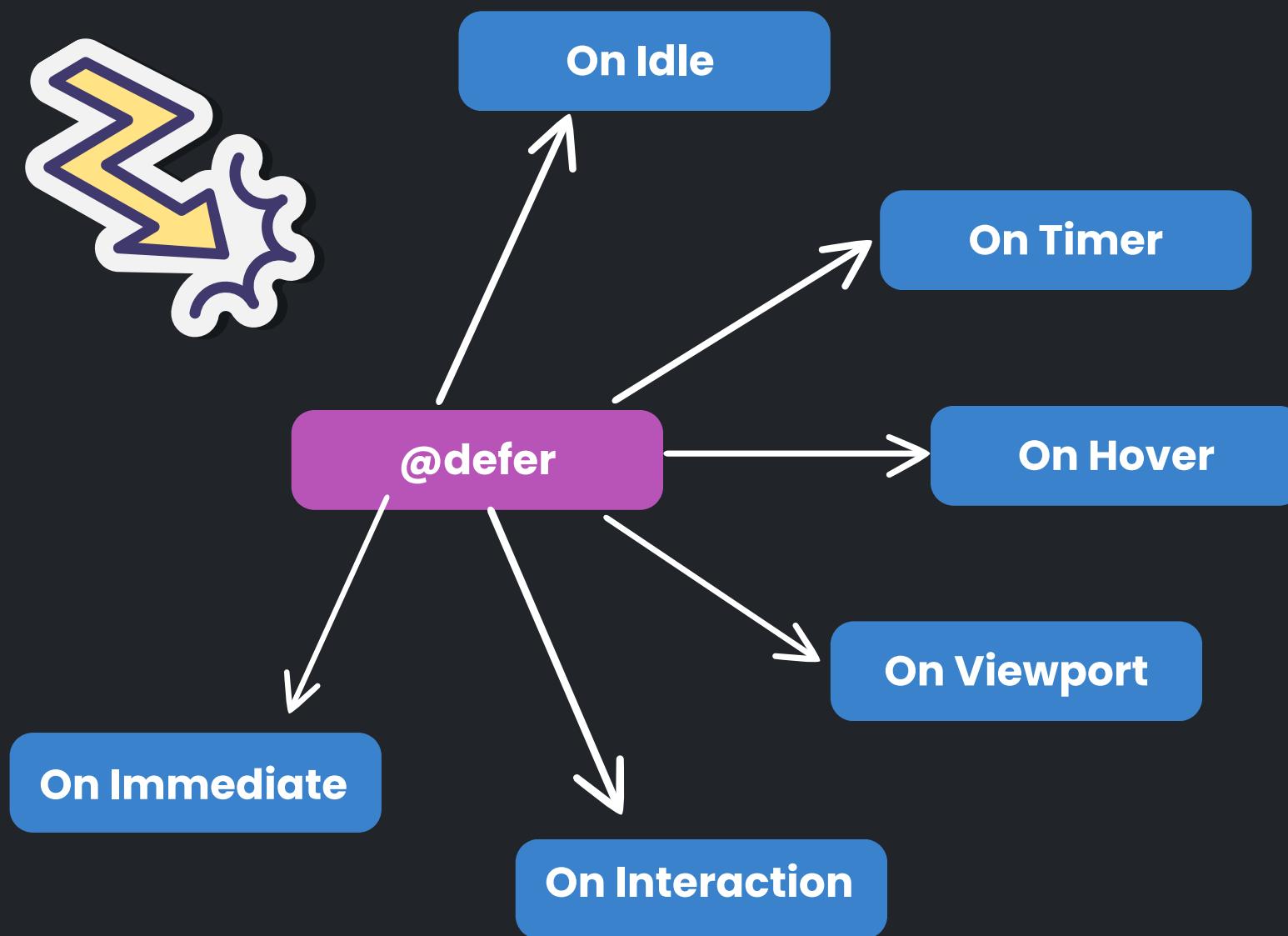
@error block : declare content that will be shown if deferred loading fails.



@coder\_aishya



# The different defer triggers available



## on idle

This is the default behaviour with a defer block. Will trigger the deferred loading once the browser has reached an idle state



@coder\_aishya



## on viewport

Loads content when the specified element enters the viewport. 

```
● ● ●  
@defer (on viewport) {  
  <chart-component />  
} @placeholder {  
  <div>Chart placeholder</div>  
}
```

## on interaction

The deferred block will be triggered upon user interaction with the specified element, either through click or keydown events.

```
● ● ●  
@defer (on interaction) {  
  <chart-component />  
} @placeholder {  
  <div>Chart placeholder</div>  
}
```



@coder\_aishya



## on hover

Hovering over the trigger area initiates deferred loading, triggered by mouseenter and focusin events.



```
@defer (on hover) {  
  <chart-component />  
} @placeholder {  
  <div>Chart placeholder</div>  
}
```

## on immediate

triggers the deferred load immediately



```
@defer (on immediate) {  
  <chart-component />  
} @placeholder {  
  <div>Chart placeholder</div>  
}
```



@coder\_aishya



## on timer

The `timer(x)` will activate after a specified duration, where ‘x’ represents the time interval. This duration is required and can be specified in milliseconds (ms) or seconds (s)



```
@defer (on timer(500ms)) {  
  <chart-component />  
}
```



@coder\_aishya



**Follow me for more**



aishwarya-dhuri



coder\_aishya