



Reusable animations

Prerequisites

A basic understanding of the following concepts:

- [Introduction to Angular animations](#)
- [Transition and triggers](#)

The [AnimationOptions](#) interface in Angular animations enables you to create animations that you can reuse across different components.

Creating reusable animations

To create a reusable animation, use the `animation()` method to define an animation in a separate `.ts` file and declare this animation definition as a `const` export variable. You can then import and reuse this animation in any of your app components using the `useAnimation()` API.

src/app/animations.ts

```
import {
  animation, trigger, animateChild, group,
  transition, animate, style, query
} from '@angular/animations';

export const transAnimation = animation([
  style({
    height: '{{ height }}',
    opacity: '{{ opacity }}',
    backgroundColor: '{{ backgroundColor }}'
  }),
  animate('{{ time }}')
]);
```

In the above code snippet, `transAnimation` is made reusable by declaring it as an export variable.

Note: The `height`, `opacity`, `backgroundColor`, and `time` inputs are replaced during runtime.

You can import the reusable `transAnimation` variable in your component class and reuse it using the `useAnimation()` method as shown below.

src/app/open-close.component.ts

```
import { Component } from '@angular/core';
import { useAnimation, transition, trigger, style, animate } from
 '@angular/animations';
import { transAnimation } from '../animations';

@Component({
  trigger('openClose', [
    transition('open => closed', [
      useAnimation(transAnimation, {
        params: {
          height: 0,
          opacity: 1,
          backgroundColor: 'red',
          time: '1s'
        }
      })
    ])
  ])
],
})
```

More on Angular animations

You may also be interested in the following:

- [Introduction to Angular animations](#)
- [Transition and triggers](#)
- [Complex animation Sequences](#)
- [Route transition animations](#)

