Angular Page Transitions

A project to demonstrate and test Angular animation transitions.

The package

To use the animations feature; you need to install the package:

```
npm install @angular/animations
```

You then need to add the '`BrowserAnimationModule`' to the '`app.module.ts`' file:

```
import { BrowserAnimationsModule } from '@angular/platform-
browser/animations';

imports: [ ... BrowserAnimationsModule
```

Note: *The '`NoopAnimationModule`' conflicts with these animations, and will stop them working.*

Use

Animations can only be applied to elements. By adding the transistion animation to a wrapping-div (which may already be in place); you can create the illusion of page transitions.

I've placed the animation within a callable function for cleanness - this is also how one of the Google examples does it.

The below needs to be done for each component that you would like to add the animation to - most likely; the components that handle route requests.

Import the function and the references

```
import { trigger, state, style, animate, transition } from
'@angular/animations';

import { fade } from "../animations/transitions";
```

Use the function to apply the animation

```
@Component({

selector: 'test-home',

templateUrl: './home.component.html',

styleUrls: ['./home.component.css'],

animations: [fade()]
```

```
})
```

Note the '`animations: [fade()]`'.

Apply the animation to the element

```
<div @fade>
```

Node the '@' sign used to depict an animation.

Notes

The animations

The animations follow the same pattern as CSS animations, with a starting state, and an end state - though Angular does not yet allow mid-points in animations (just a start and a finish).

The animation goes inside the Component decoration.

Example block:

```
trigger('fade', [

state('in', style({ opacity: 0 })),

transition(':enter', [

style({ opacity: 0 }),

animate(500, style({ opacity: 1 }))

]),

transition(':leave', [

style({ opacity: 1 }),

animate(500, style({ opacity: 0 }))

])

])
```

Syntax

Give the animation an alias

```
trigger('fade', [
```

Here the alias for the animation is 'fade'. Aliases are used for applying animations to other elements within templates, too.

Initial state before animations start

```
state('in', style({ opacity: 0 })),
```

The 'state' is the default set of styles that the element (or page) should have before the animation(s) start.

'in' refers to the fact that these are styles that should be applied before the animation(s) start - preparing to go 'in' to the animation

Transition

```
transition(':enter', [

style({ opacity: 0 }),

animate(500, style({ opacity: 1 }))

]),
```

This is defining the end-point styles that should be animated to, from the 'state'/initial styles.

First you pass the condition that needs to be met before the animation will fire. The page state is 'void' before it is being used, and '*' is a wildcard for any other state - this animation will fire if the page is moving from unused, to anything else.

The common 'fade-in', 'fade-out', states have aliases of thier own (used in the example):

```
transition(':enter', [ ... ]); // void => *

transition(':leave', [ ... ]); // * => void
```

Second; you pass the transition. Here we animate the styles, passing an object containing CSS-style notations, and then set the time that the animation should take to finish to '500' (miliseconds).

```
style({ opacity: 0 }),

animate(500, style({ opacity: 1 }))
```

This animation will cause the page element to change styles from 'opacity: 0' (set in the 'state' section, and then repeated in the 'style' section of the transition), to 'opacity: 1'.

This page will fade in, finishing in 500 miniseconds, when the page is entered, and back out again when leaving.

Timings

Animation duration can be set using the following:

- As a plain number, in milliseconds: 100
- In a string, as milliseconds: '100ms'
- In a string, as seconds: '0.1s'

The function itself

The file that the function is in is very simple - exporting the above, and not much more:

```
import { trigger, state, style, animate, transition } from
'@angular/animations';

export function fade(timeSpan: number = 1000): any {

return trigger('fade', [

state('in', style({ opacity: 0 })),

transition(':enter', [

style({ opacity: 0 }),

animate(timeSpan, style({ opacity: 1 }))` ` ]),``

transition(':leave', [

style({ opacity: 1 }),

animate(timeSpan, style({ opacity: 0 }))

])

]);

}
```

References

Angular documentation: https://angular.io/guide/animations

Google Developer Experts: https://medium.com/google-developer-experts/angular-2-animate-router-transitions-6de179e00204