

# Class 26 - Angular Animations & Adding Offline Capabilities with Service Workers

---

## [Class 26 Course Content](#)

### Lesson Outline

Today we will learn:

1. **How to add Animations to Angular Applications**
  2. **How to create a service worker**
  3. **How to add offline capabilities**
- 
- 

### Lesson Notes

- **PWA:** A *Progressive Web App (PWA)* is a web application that has the feel and functionality of an actual native/mobile phone app. PWA's are also enhanced with modern API's to deliver advanced features such as the ability for a user to install the website on any device.
  - **Service Worker:** A *Service Worker* is a script that the browser downloads and runs in the background, separate from the main thread of the webpage, opening the door to features that don't need to work with the webpage or user interaction. These features include, but are not limited to, *push notifications*, *background sync*, and supporting offline experiences!
- 
- 

### Animations - Course Project Steps

#### STEP 1: Setting up Angular for Animations

*app.module.ts* file:

- Import `{BrowserAnimationsModule}` from `"@angular/platform-browser/animations"`.
- Add the `"BrowserAnimationsModule"` to the `"imports"` array.

```
import { BrowserAnimationsModule } from '@angular/platform-  
browser/animations';  
  
// . . .  
  
imports: [  
  // . . .  
  BrowserAnimationsModule,  
],
```

## STEP 2: Adding a Swoop Animation

*library/book-results/book-results.component.ts file:*

- Add the "animations" array to the `@Component({})` decorator config.
- Create a new `trigger()` that is called "swoopIn".
- This animation should use the state "in" to have the element eventually be the normal size and full opacity.
- Add the `transition()` method that listens for the ":enter" event and animates using keyframes the "transform", "opacity", and "offset" properties from 0 to 1.
- **NOTE:** Be sure to import `{ animate, keyframes, state, style, transition, trigger }` from `"@angular/animations"`.

```
animations: [  
  trigger('swoopIn', [  
    state(  
      'in',  
      style({  
        transform: 'scale(1)',  
        opacity: 1,  
      })  
    ),  
    transition(':enter', [  
      animate(  
        '1s',  
        keyframes([  
          style({  
            transform: 'scale(0)',  
            opacity: 0,  
            offset: 0,  
          }),  
          style({  
            transform: 'scale(.3)',  
            opacity: 0.3,  
            offset: 0.2,  
          }),  
          style({  
            transform: 'scale(.6)',  
            opacity: 0.6,  
            offset: 0.4,  
          }),  
          style({  
            transform: 'scale(.9)',  
            opacity: 0.9,  
            offset: 0.8,  
          }),  
          style({  
            transform: 'scale(1)',  
            opacity: 1,  
          })  
        ])  
      )  
    ]  
  )  
])
```

```

        offset: 1,
      })),
    ]),
  ),
  1),
  1),
  1),
],

```

*library/book-results/book-results.component.html:*

- Inside the opening tag of the outer-most `<div>`, add the binding for the "swoopIn" animation.

```

<div class="row mb-3" *ngIf="bookResults.length > 0" [@swoopIn]>
  <!-- . . . -->
</div>

```

## Service Workers & PWA - Course Project Steps

### STEP 1: Setting up Angular PWA Functionality

*Terminal:*

- Use the Angular CLI to add `@angular/pwa` to the project.
- Globally Install a lightweight node server by running `npm i -g http-server`.
- Build the project for production.
- Change directories into the "dist/PROJECTNAME/browser" folder
- Run "http-server".

```

ng add @angular/pwa
# . . .
npm i -g http-server
# . . .
ng build
# . . .
cd dist/PROJECTNAME/browser
# . . .
http-server

```

- **NOTE:** To see if it installed, refresh the page and check the "Application" tab in developer tools and select "Service Workers".

## Additional Notes

### Resources

- [Angular Docs - Getting Started w/ Service Workers](#)
- [Blog Article - What is a PWA](#)