Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

Support us   .Net Basics   C# SQL   ASP.NET   ADO.NET   MVC   Slides   C# Programs   Subscribe   Buy DVD

Angular route loading indicator

Suggested Videos

Part 53 - Create observable from array | Text | Slides
Part 54 - Angular resolve guard | Text | Slides
Part 55 - Angular router navigation events | Text | Slides

In this video we will discuss **how to display a loading indicator** if there is a delay when navigating from one route to another route in an angular application. This is continuation to Part 55. Please watch Part 55 from Angular CRUD tutorial before proceeding.

At the moment in our application, when we navigate to the **LIST** route, it will take 2 seconds to pre-fetch data required for the **LIST** route. This is because, we have a route resolver configured on the **LIST** route. We implemented this route resolver in Part 54 of Angular CRUD tutorial.

During the 2 seconds wait time, while the route resolve guard is busy retrieving the required data, we want to display a loading indicator, so the user knows the application is busy processing the request and he does not end up clicking on the link multiple times.
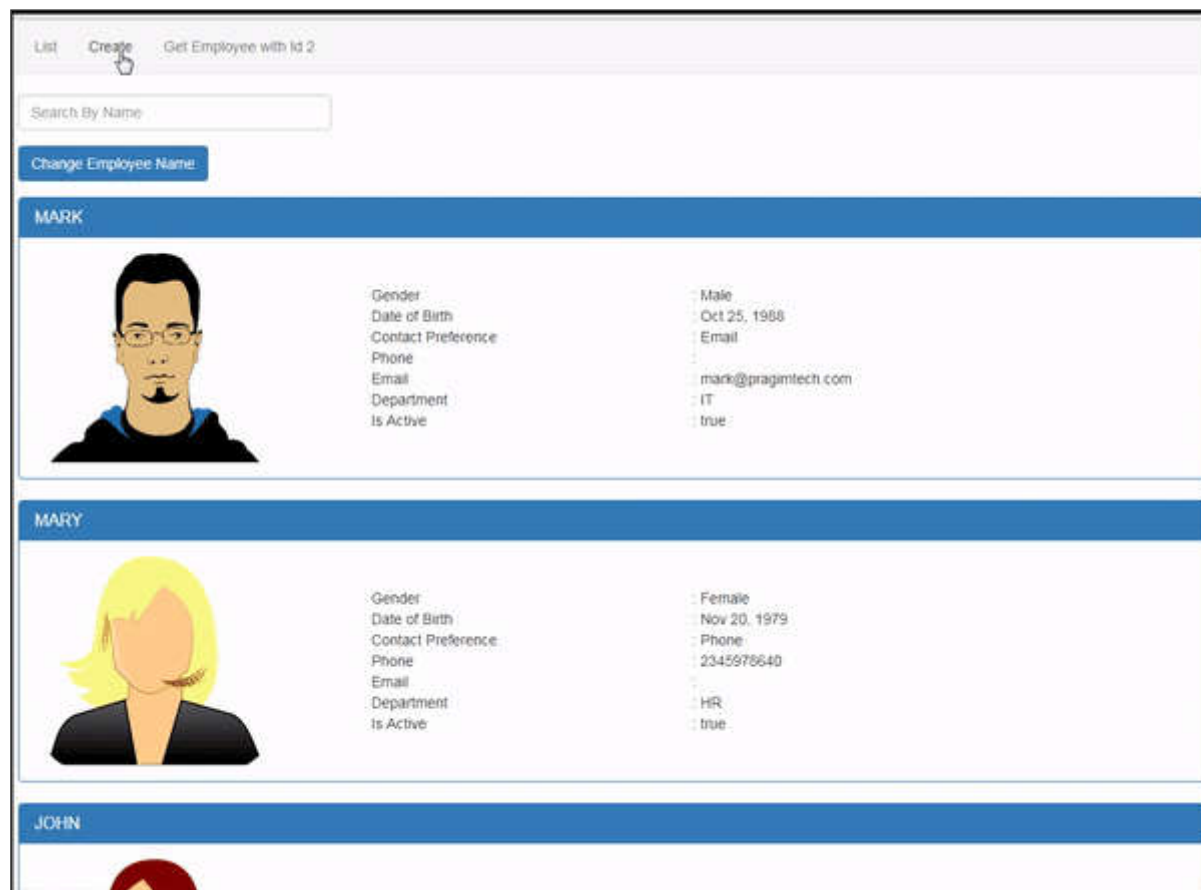
Complete Tutorials

JavaScript tutorial

Bootstrap tutorial

Angular tutorial for beginners

Angular 5 Tutorial for beginners

Important Videos

The Gift of Education

Web application for your business

To implement the loading indicator, we are going to make use of the **Angular Router Navigation events**. We discussed these events in our previous video. These navigation events range from when the navigation starts and ends to many points in between. When the navigation starts, we want to show the loading indicator, and when the navigation ends, hide the loading indicator.

To be able to react and execute some code in response to the router navigation events, subscribe to the Angular router events observable.

Step 1 : Modify the code in Root Component (AppComponent) in app.component.ts as shown below.

```
import { Component, } from '@angular/core';
// Import the Router and navigation events
import {
  Router, NavigationStart, NavigationEnd,
  NavigationCancel, NavigationError, Event
} from '@angular/router';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app';
  // We will use this property to show or hide
  // the loading indicator
  showLoadingIndicator = true;

  // Inject the Angular Router
  constructor(private _router: Router) {
    // Subscribe to the router events observable
    this._router.events.subscribe((routerEvent: Event) => {

      // On NavigationStart, set showLoadingIndicator to ture
      if (routerEvent instanceof NavigationStart) {
        this.showLoadingIndicator = true;
      }

      // On NavigationEnd or NavigationError or NavigationCancel
      // set showLoadingIndicator to false
      if (routerEvent instanceof NavigationEnd ||
        routerEvent instanceof NavigationError ||
        routerEvent instanceof NavigationCancel) {
        this.showLoadingIndicator = false;
      }

    });
  }
```

```
}
```

Step 2 : Bind to the **showLoadingIndicator** property in the view template of our root component i.e AppComponent in app.component.html file.

```html
<div class="container">
    <nav class="navbar navbar-default">
        <ul class="nav navbar-nav">
            <li>
                <a routerLink="list" queryParamsHandling="preserve">List</a>
            </li>
            <li>
                <a routerLink="create">Create</a>
            </li>
            <li>
                <a [routerLink]="['employees',2]">
                    Get Employee with Id 2
                </a>
            </li>
        </ul>
    </nav>
    <router-outlet>
    </router-outlet>
    <!-- Bind to showLoadingIndicator property in the component class -->
    <div *ngIf="showLoadingIndicator" class="spinner"></div>
</div>
```

Step 3 : We are using CSS animations to get the effect of a loading spinner. Place the following CSS in app.component.css file

```css
.spinner {
  border: 16px solid silver;
  border-top: 16px solid #337AB7;
  border-radius: 50%;
  width: 80px;
  height: 80px;
  animation: spin 700ms linear infinite;
```

```
    top:50%;
    left:50%;
    position: absolute;
  }
 @keyframes spin {
   0% { transform: rotate(0deg) }
   100% { transform: rotate(-360deg) }
 }
```

The following website has different loading spinners.
https://loading.io/css/

We have to slightly modify some of the CSS properties to be able to use them on our list page. I made the following changes.

- Removed *display* property
- Changed *position* property value from relative to fixed
- Add *top* and *left* properties and set them to 50%
- Changed background property value to *#337AB7*;

With the above changes I have a spinner as shown below.

List   Create   Get Employee with Id 2

Search By Name

Change Employer Name

MARK

Gender	Male
Date of Birth	Oct 25, 1988
Contact Preference	Email
Phone	
Email	mark@pragimtech.com
Department	IT
Is Active	true

MARY

Gender	Female
Date of Birth	Nov 20, 1979
Contact Preference	Phone
Phone	2345979640
Email	
Department	HR
Is Active	true

JOHN

2 comments:

attu June 9, 2018 at 4:08 PM

Dear Sir ..Kindly tell me about routes.
i mean when i login redirect to home
there 3 nav bars admin panel .account,manufacture.
when i click on one of them .my navbars hide and i want navbar remain .. what should i do

Reply

Abhinav Oruganti November 15, 2018 at 4:15 AM

hi sir you are binding the spinner for links but i need it for events, could you please provide me an example if possible

Reply

```
Enter your comment...
```

Comment as: toilati123vn@g ▼

Publish   **Preview**

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Links to this post

Create a Link

Newer Post   Home   Older Post

Subscribe to: Post Comments (Atom)

Powered by Blogger.