Angular 16 : Online Class

sahosoft

Online Learning Platform

Routing

By: Sahosoft Solutions

Presented by : Chandan Kumar

Angular Routing

- Routing is mechanism used by angular for navigating between page and displaying appropriated component/page on browser.

- Angular framework is mainly focused and good for the SPA. It loads a single full HTML page and dynamically loads or updates the partial pages as per user request. And, that is achieved with the help of router. A set of partial page loading rule and urls is defined in router to render or load partial pages based on user request.

- Angular routing helps navigation across the application from one view to another view. It also allows us to maintain the state, implement modules, and load the modules based on the role of the user.

Angular Routing

- In component based architecture of Angular 2 + , Angular looks into the routes array, **match the path** as per the route requested and loads the component relevant to the requested route as well as makes available the relevant data for that particular route.

- Angular provides an easy way to create and work with components, in a single page application(SPA) it is essential to work with multiple views/screens, navigate and communicate between them. Angular provides router service to manage this in a very easy way.

- An Angular application that uses Angular Router only has one router service instance: **It's a singleton**. Whenever and wherever you inject the Router service in your application, you'll get access to the same Angular Router service instance

Why Routing

- We access our application through one URL such as http://localhost:4200 and our application is not aware of any other URLs such as http://localhost:4200/Login
- Most web applications need to support different URLs to navigate users to different pages in the application. That's where a router comes in.

Angular Router

- **Angular Router** is an official Angular routing library, written and maintained by the Angular Core Team.

- It's a JavaScript router implementation that's designed to work with Angular and is packaged as @angular/router.

- First of all, Angular Router takes care of the duties of a JavaScript router:

- it activates all required Angular components to compose a page when a user navigates to a certain URL

- it lets users navigate from one page to another without page reload

- it updates the browser's history so the user can use the back and forward buttons when navigating back and forth between pages.

Angular Router

- **Angular Router allows us to:**
- redirect a URL to another URL
- resolve data before a page is displayed
- run scripts when a page is activated or deactivated
- lazy load parts of our application.

How Angular Router Works

When a user navigates to a page, Angular Router performs the following steps in order:
Every time a link is clicked or the browser URL changes, Angular router makes sure your application reacts accordingly.

To accomplish that, Angular router performs the following 7 steps in order:

Step 1 - Parse the URL:
In step 1 of the routing process, Angular router takes the browser URL and parses it as a URL tree.
A URL tree is a data structure that will later help Angular router identify the router state tree in step 3

How Angular Router Works

- To parse the URL, Angular uses the following conventions:

- / - slashes divide URL segments
- () - parentheses specify secondary routes
- : - a colon specifies a named router outlet
- ? - a question mark separates the query string parameters
- // - a double slash separates **multiple secondary routes**

How Angular Router Works

Step 2 – Redirect

- Before Angular router uses the URL tree to create a router state, it checks to see if any redirects should be applied.
- **Only one redirect is applied!**

How Angular Router Works

Step 3 - Identify the router state

- Angular router traverses the URL tree and matches the URL segments against the paths configured in the router configuration.
- If a URL segment matches the path of a route, the route's child routes are matched against the remaining URL segments until all URL segments are matched.
- If no complete match is found, the router backtracks to find a match in the next sibling route

How Angular Router Works

Step 4 - Guard - run guards

- At the moment, *any* user can navigate *anywhere* in the application *anytime*. That's not always the right thing to do.
- Perhaps the user is not authorized to navigate to the target component.
- Maybe the user must login (*authenticate*) first.
- Maybe you should fetch some data before you display the target component.
- You might want to save pending changes before leaving a component.
- You might ask the user if it's OK to discard pending changes rather than save them.
- **You can add *guards* to the route configuration to handle these scenarios.**

How Angular Router Works

- **Step 5 - Resolve - run resolvers**
- it resolves the required data for the router state.

- **Step 6- Activate**
- it activates the Angular components to display the page.

- **Step 7 – Manage**
- Finally, when the new router state has been displayed to the screen, Angular router listens for URL changes and state changes.
- it manages navigation and repeats the process when a new URL is requested.