**What is Angular Router?**

An Angular Router is **a tool**, library that **configures navigations between states and views** within your Angular app.

The Routing library is written and **maintained by the**[**Angular Core Team**](https://angular.io/about?group=Angular)**.**

Angular router has own library package - @angular/router.

import {Routes, RouterModule,}  from '@angular/router';

**The basic concept of Angular Router** and It allows you to -

1.      Redirect a URL to another URL

2.      Resolve data before a page is displayed

3.      Run scripts when a page is activated or deactivated

4.      Lazy load parts of our application

The router supports both styles with **two LocationStrategy** providers -

1.      **PathLocationStrategy**— this is the **default style**.

2.      **HashLocationStrategy**— **adds the route path to the hash (#)** in the browser’s URL.

**What is Router module?**

The Router module is a **module** that provides the necessary service providers and directives for navigating one view to other in the application.

**What is Routes?**

Angular Route is **an array of route configurations**. The “RouterModule.forRoot” method in the module imports to configure the router.

type Routes = Route[];

Each Route has the following properties -

interface Route {

  path?: string

  pathMatch?: string

  matcher?: UrlMatcher

  component?: Type<any>

  redirectTo?: string

  outlet?: string

  canActivate?: any[]

  canActivateChild?: any[]

  canDeactivate?: any[]

  canLoad?: any[]

  data?: Data

  resolve?: ResolveData

  children?: Routes

  loadChildren?: LoadChildren

  runGuardsAndResolvers?: RunGuardsAndResolvers

}

List of properties and it has the following order -

1.      **path** - It uses the route matcher DSL

2.      **pathMatch** - It uses to specifies the matching strategy

3.      **matcher** - It uses to defines a custom strategy for path matching

4.      **component** - It is a component type

5.      **redirectTo** - It is the URL fragment and it will replace the current matched segment

6.      **outlet** - It is the name of the outlet the component should be placed into

7.      **canActivate**  - It is an array of DI tokens  and used to handle the CanActivate handlers

8.      **canActivateChild** - It is an array of DI tokens and used to handle the CanActivateChild handlers

9.      **canDeactivate** - It is an array of DI tokens and used to handle the CanDeactivate handlers

10.  **canLoad** - It is an array of DI tokens and used to handle the CanLoad handlers

11.  **data** - It is additional data provided to the component by using the ActivatedRoute

12.  **resolve** - It is a map of DI tokens used to look up data resolvers

13.  **runGuardsAndResolvers** - It is defined when guards and resolvers will be run and by default, they run only when the matrix parameters of the route change.

14.  **children** - it is an array of child route definitions

15.  **loadChildren** - It is a reference to lazily loaded child routes.