



Angular routing

GFT INTERNAL TRAINING

INNOVATE. TRANSFORM. DELIVER.

Router

- Angular Router is an optional service that presents a particular view (set of components) for a given URL
- it is not part of the Angular core. It is in its own library package, @angular/router
- to use it import what you need from it

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


Agenda

- 1. Adding Angular Router**
- 2. Configuring Routes**
- 3. Navigation**
- 4. Route parameters**
- 5. Angular router overview**
- 6. Learning resources**

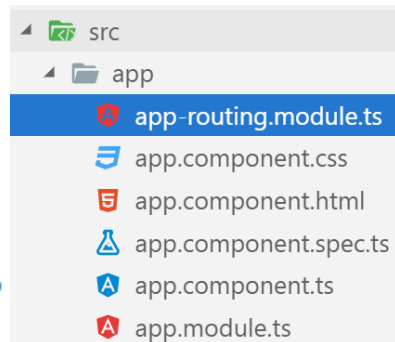
Adding Angular Router

- since angular cli 7, you can add the boilerplate code needed to use angular router
- this code already worked in previous versions

```
C:\[redacted]>ng new withRouter  
? Would you like to add Angular routing? (y/N) y
```


 app-routing.module.ts ✕

```
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3
4 const routes: Routes = [];
5
6 @NgModule({
7   imports: [RouterModule.forRoot(routes)],
8   exports: [RouterModule]
9 })
10 export class AppRoutingModule { }
```



app.module.ts ✕



```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { AppRoutingModule } from './app-routing.module';
4 import { AppComponent } from './app.component';
5 @NgModule({
6   declarations: [
7     AppComponent
8   ],
9   imports: [
10    BrowserModule,
11    AppRoutingModule
12  ],
```

 app.component.html ✕

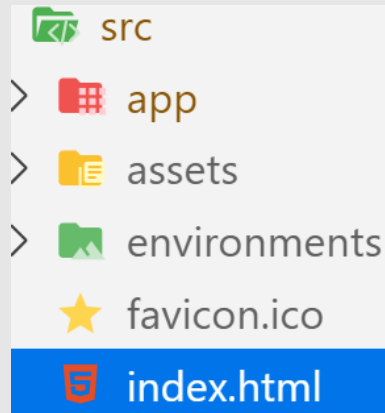
```
<h2>Here are some links to help you start: </h2>
<ul>
  <li>
    <h2><a target="_blank" rel="noopener" href="...">...</h2>
  </li>
  <li>
    <h2><a target="_blank" rel="noopener" href="...">...</h2>
  </li>
  <li>
    <h2><a target="_blank" rel="noopener" href="...">...</h2>
  </li>
</ul>

<router-outlet></router-outlet>
```

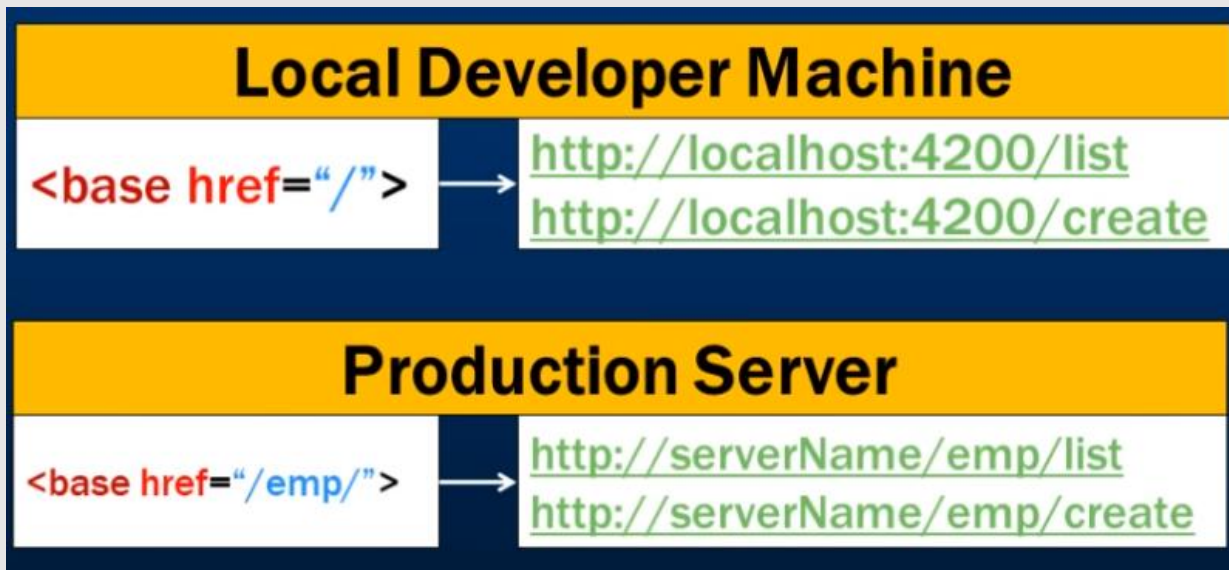

Base URL Tag

The Base URL tag is set within the <head> tag of index.html so the path are relative to the app (in development typically http://localhost:4200)

```
<html>
<head>
  <meta charset="utf-8">
  <title>Prueba1</title>
  <base href="/">
  <meta name="viewport" content="width
  <link rel="icon" type="image/x-icon"
</head>
```



Base URL Tag



Good angular href explanation <https://www.youtube.com/watch?v=Std1QJpMEiE>
Since Angular 7 → `ng build --prod --base-href /emp/`

Configuring Routes

Routes definition object

Routes type -> array of routes that defines the routing for the application

```
const routes: Routes = [  
  { path: '', redirectTo: '/about', pathMatch: 'full' },  
  { path: 'about', component: AboutComponent },  
  { path: 'list', component: ListComponent },  
  { path: 'add', component: AddComponent },  
  { path: 'detail/:id', component: DetailComponent },  
  { path: '**', component: NotFoundComponent }  
];
```

Route properties

- **path** -> URL to be shown in the browser when app is on this route
- **component** -> component to be rendered when the app is on this route
- **redirectTo** -> redirect route
- **pathMatch** -> defaults to 'prefix'; determines whether to match full URLs or just the beginning. When defining a route with "" set pathMatch to 'full', otherwise it will match all paths
- **children** - array of route definitions objects representing the child routes of this route
- ...

<https://angular.io/api/router/Route#properties>

Routes definition object

- each route has a path
- each route has a component or a redirect
- order of the routes in the configuration matters, because the router uses a first-match wins strategy when matching routes => more **specific routes should be placed above** less specific routes
- therefore the wildcard will be always the last because it matches every URL and should be selected only if no other route has been matched first

```
{ path: 'add', component: AddComponent },  
{ path: 'detail/:id', component: DetailComponent },  
{ path: '**', component: NotFoundComponent }
```


Router outlet

where the components of the routes will be render

```
<router-outlet></router-outlet>
```

Navigation

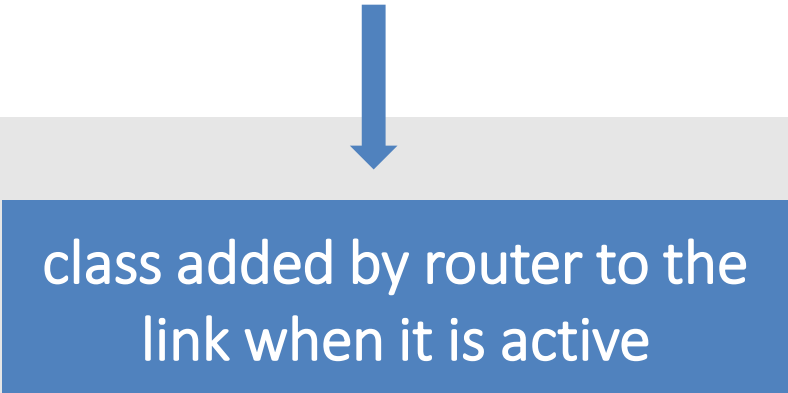
Navigation with links

```
<nav>
  <ul>
    <li><a routerLink="/home" routerLinkActive="active">Home</a></li>
    <li><a routerLink="/login" routerLinkActive="active">Login</a></li>
    <li><a routerLink="/about" routerLinkActive="active">About us</a></li>
  </ul>
</nav>
```



path

A blue box labeled 'path' with a blue arrow pointing down to it from the 'routerLink' attribute in the code above.



class added by router to the
link when it is active

A blue box containing the text 'class added by router to the link when it is active' with a blue arrow pointing down to it from the 'routerLinkActive' attribute in the code above.

Navigation programmatically

```
import { Router } from '@angular/router';
```

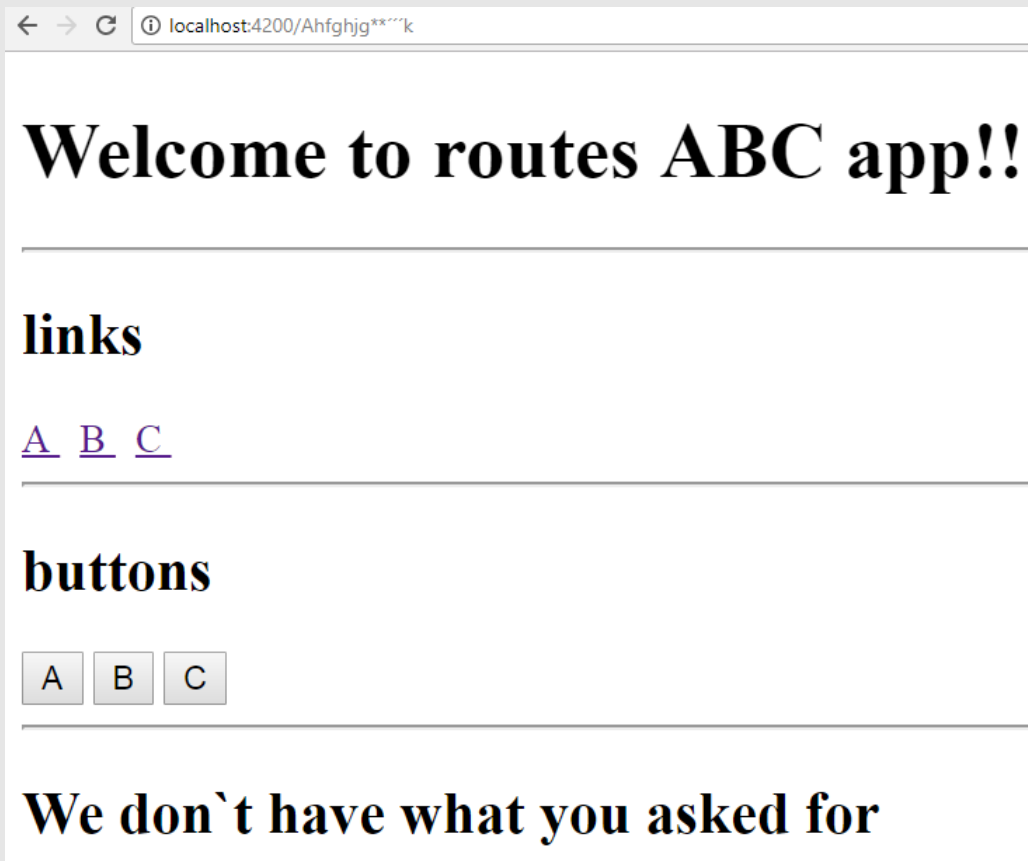
```
constructor( private cancionesService: CancionesService,  
             private router: Router) { }
```

```
this.router.navigate(['list']);
```

learn by doing

- create an angular app with angular-cli
- add these components: links, buttons, A, B and C
- links and buttons will always appear
- links will show links to A, B and C
- buttons will show buttons to programmatically go to A, B, C
- put the routes definitions in a const in app-routing.module.ts
- the link to the active state should be highlighted
- if we request localhost:4200 it will show A component
- a component with an error message should appear if we request a non existing URI





Routes parameters

Declaring Route Parameters

```
{ path: '', redirectTo: 'product-list', pathMatch: 'full' },  
{ path: 'product-list', component: ProductList },  
{ path: 'product-details/:id', component: ProductDetails }
```

For example, to see the product details page for product with id 5, you should use the following URL:

`localhost:4200/product-details/5`

Navigation with parameters

- with links

```
<a *ngFor="let product of products"  
  [routerLink]="['/product-details', product.id]">  
  {{ product.name }}  
</a>
```

- programmatically

```
goToProductDetails(id) {  
  this.router.navigate(['/product-details', id]);  
}
```

Reading route parameters

```
export class LoanDetailsPage implements OnInit, OnDestroy {  
  id: number;  
  private sub: any;  
  
  constructor(private route: ActivatedRoute) {}  
  
  ngOnInit() {  
    this.sub = this.route.params.subscribe(params => {  
      this.id = +params['id']; // (+) converts string 'id' to a number  
  
      // In a real app: dispatch action to load the details here.  
    });  
  }  
  
  ngOnDestroy() {  
    this.sub.unsubscribe();  
  }  
}
```

params → an Observable that contains the required and optional parameters specific to the route
Works in Angular 8 but it is deprecated
use paramMap instead

Reading route parameters

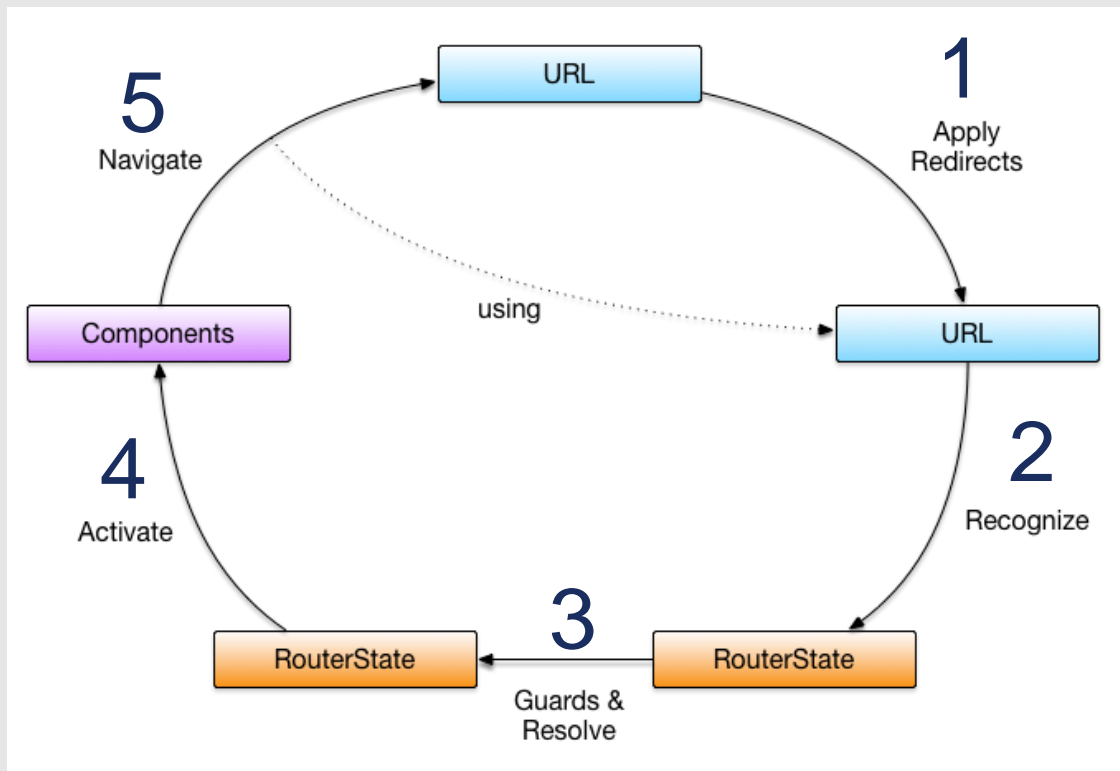
- params property on ActivatedRoute is an Observable
- this allows not recreate the component when navigating to the same component
- in this case the parameter may change without the component being recreated

Angular router overview

Some concepts

- **router state** -> arrangement of application components that defines what is visible on the screen
- **navigation** -> transition from one router state to another
- **router job** -> manage navigation
- **URL** (in a well managed web app) -> serialized router state

Angular router



<https://leanpub.com/router>

Learning resources

Learning resources

- <https://angular.io/guide/router>
- <https://angular.io/guide/router#the-sample-application>
- book by Angular Router main contributor Victor Savkin
<https://leanpub.com/router> (free sample)



Shaping the future of digital business

GFT Internal Technical Training

Eduardo García Ibaseta

eduardo.garcia-ibaseta@gft.com
+34 935 639476

GFT IT Consulting, S.L.

Av. Alcalde Barnils, 69-71

08174 Sant Cugat del Vallès (BARCELONA)