

2-SPA

Páginas y rutas Angular SPA

- 1. Rutas**
- 2. Lazy Loading**
- 3. Rutas anidadas**
- 4. Parámetros**

1. Rutas

RouterModule

Router Outlet

Router Link

1.1 RouterModule

`AppRoutingModule` importa, configura y exportar al `RouterModule`

```
import { Routes, RouterModule } from '@angular/router';
const routes: Routes = [
  {
    path: 'heroes',
    component: HeroesComponent
  },
  {
    path: 'not-found',
    component: NotFoundComponent
  },
  {
    path: '**',
    redirectTo: 'not-found'
  }
];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
```

Componentes y rutas

```
ng g c heroes  
ng g c core/not-found
```

El componente `HeroesComponent` se asocia con la ruta `'heroes'` El componente `NotFoundComponent` se asocia con la ruta `'not-found'`

RedirectTo

Nadie va voluntariamente a esa ruta

Sólo los que se pierden

```
{  
  path: '**',  
  redirectTo: 'not-found'  
}
```

1.2 Router Outlet

El contenido de `main.component.ts`, ahora será dinámico

```
<main class="container">  
  <router-outlet></router-outlet>  
  <!-- Dynamic content here! -->  
</main>
```

Por ejemplo el contenido de `NotFoundComponent` será

```
<h1>Not Found</h1>  
<h2>404</h2>  
<a routerLink="/">Go home</a>
```

1.2 Router Link

```
<a routerLink="/">Go home</a>
```

Es una *Directiva*

Como un atributo, pero con superpoderes

Por ahora, *simplemente* mantiene la gestión de las rutas en el lado del navegador.

Menu header

El componente `app-header` queda así:

```
<header class="sticky">
  <a routerLink="/" class="logo">
    <span class="icon-home"></span> <span>2-SPA</span>
  </a>
  <a routerLink="heroes" routerLinkActive="router-link-active" class="button">
    <span> Heroes</span>
  </a>
</header>
```


| Recap:

1. Rutas

RouterModule

Router Outlet

Router Link

2 Lazy Loading

Webpack y los bundles por ruta

El enrutador delegado

Navegación

2.1 Webpack y los bundles por ruta

- Objetivo: diferir la descarga de las rutas no visitadas
- Empaquetar cada ruta en un *bundle*
- Requiere un módulo por ruta
- Y un convenio especial con *webpack*

Crear los componentes en módulos con enrutado

```
ng g m home --routing true
ng g c home/home
ng g m about --routing true
ng g c about/about
```

Y se configuran las **rutas** con **'rutas al módulo'**

```
const routes: Routes = [
  {
    path: '',
    loadChildren: './home/home.module#HomeModule'
  },
  {
    path: 'about',
    loadChildren: './about/about.module#AboutModule'
  }
];
```

2.2 El enrutador delegado

- `loadChildren` delega el enrutado en otro módulo; `HomeRoutingModule`

```
const routes: Routes = [  
  {  
    path: '',  
    component: HomeComponent  
  }  
];
```

■ Ojo al path. En `AboutRoutingModule` sería:

```
const routes: Routes = [  
  {  
    path: '',  
    component: AboutComponent  
  }  
];
```

Comprobar en ejecución

Los bundles se descargan al navegar por las rutas

Agregamos un enlace destacado a esta nueva ruta en el HeaderComponent

```
<a routerLink="about" routerLinkActive="router-link-active" class="button">  
    
  <span> About us</span>  
</a>
```

Recap:

2 Lazy Loading

Webpack y los bundles por ruta

El enrutador delegado

Navegación

3. Rutas anidadas

Children

RouterOutlet anidado

3.1 Children

En HeaderComponent

```
<header class="sticky">
  <a routerLink="/" class="logo">
    <span class="icon-home"></span> <span>2-SPA</span>
  </a>
  <a routerLink="about" routerLinkActive="router-link-active" class="but
    <span> About us</span>
  </a>
</header>
```

```
ng g c about/about/links
ng g c about/about/info
```

En `about-routing.module.ts`

```
const routes: Routes = [
  {
    path: '',
    component: AboutComponent,
    children: [
      {
        path: 'links',
        component: LinksComponent
      },
      {
        path: 'info',
        component: InfoComponent
      }
    ]
  }
]
```

3.2 RouterOutlet anidado

En `AboutComponent` :

```
<header class="sticky">
  <a routerLink="links" class="button"> <span> Tutorial Links</span> </a>
  <a routerLink="info" class="button"> <span> More Info</span> </a>
</header>
<router-outlet></router-outlet>
```

| Recap:

3. Rutas anidadas

Children

RouterOutlet anidado

4. Parámetros

Variables en la ruta

ActivatedRoute

4.1 Variables en la ruta

Dada esta estructura nuevos components

```
ng g c about/about/authors  
ng g c about/about/authors/author
```

Podemos gestionar dichas rutas en `about-routing.module.ts`

```
{
  path: '', component: AboutComponent,
  children: [
    {
      path: 'links', component: LinksComponent
    },
    {
      path: 'info', component: InfoComponent
    },
    {
      path: 'authors', component: AuthorsComponent
    },
    {
      path: 'authors/:id', component: AuthorComponent
    }
  ]
}
```

Resuelve rutas como: */authors/albertobasalo* o */authors/johndoe*

4.2 ActivatedRoute

Contenido del fichero `author.component.ts` relacionado con la obtención del parámetro de la ruta activa:

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';

export class AuthorComponent implements OnInit {
  public authorId = '';
  constructor(activateRoute: ActivatedRoute) {
    this.authorId = activateRoute.snapshot.params['id'];
  }
  ngOnInit() {}
}
```

Usamos la instancia `activateRoute` de la clase `ActivatedRoute`

Para acceder a datos de la URL activa y mostrar los datos en la vista

Enlazamos todo agregando una entrada en `AboutComponent` :

```
<a routerLink="authors" class="button"> <span> Credit Authors</span> </a>
```

Y en `authors.component.html`

```
<a routerLink="albertobasalo" class="button"> <span> Alberto Basalo</span> </a>  
<a routerLink="johndoe" class="button"> <span> John Doe</span> </a>
```

| Recap:

4. Parámetros

Variables en la ruta

ActivatedRoute

| Next:

Formularios, tablas y modelos de datos en Angular

To be defined

