Router Outlet

A continuación, voy a explicar los pasos a seguir para utilizar el **Router Outlet** en Angular ilustradas con capturas con el contenido de los distintos archivos y componentes.

app.routes.ts

Una vez hayamos creado los componentes, debemos configurar aquellos que vayan a ser **rutas llamables** (es decir, los que vayan a cambiar dentro de la etiqueta <router-outlet> en el html) desde app.routes.ts de la siguiente forma:

1. Añadimos el **import con el nombre del componente** (el que esté definido en la función “export” del componente a importar, como en este caso AboutComponent, TopsComponent, etc) y **su ruta**. En este caso, el componente “about” se encuentra directamente en el directorio app, al igual que app.routes.ts, mientras que los que corresponden a los productos están dentro de un directorio llamado “products”, por lo que hay tenerlo en cuenta para definir la ruta.
2. En en el “export” de app.routes.ts, definimos las rutas (‘path’), indicando cuál es **el nombre con el que llamaremos a este componente** desde los html (ejemplo: ‘about’), y relacionándolo con el componente importado arriba (AboutComponent).
3. Para **definir una ruta por defecto**, es decir, la que se cargará cuando carguemos la web sin especificarle una ruta concreta (path: ’’), lo definimos como se ve en la línea 8 de la siguiente imagen, indicándole **a qué path hará la redirección** (redirectTo). También podríamos indicarle una redirección de un path a otro.

app.component.ts

Para poder utilizar tanto **los componentes que hemos definido en app.routes.ts**, así como los fijos (Header, Footer), debemos hacer las **importaciones necesarias desde el archivo** .ts correspondiente. En este caso, desde el de **app.component.ts**.

Los componentes fijos deben **importarse directamente** (líneas 3 y 4).

Para el contenido variable (los definidos en app.routes.ts), debemos **importar RouterOutlet** (línea 2) desde el .ts del **componente desde el que se mostrarán** (se mostrarán, en este caso, en app.component.html), y RouterLink en el .ts de los componentes desde donde se llamarán (ver siguiente punto).

Todos **los imports** que añadamos deben **incluirse en la línea "imports"** dentro de @component (como se muestra en la línea 10 en esta imagen):

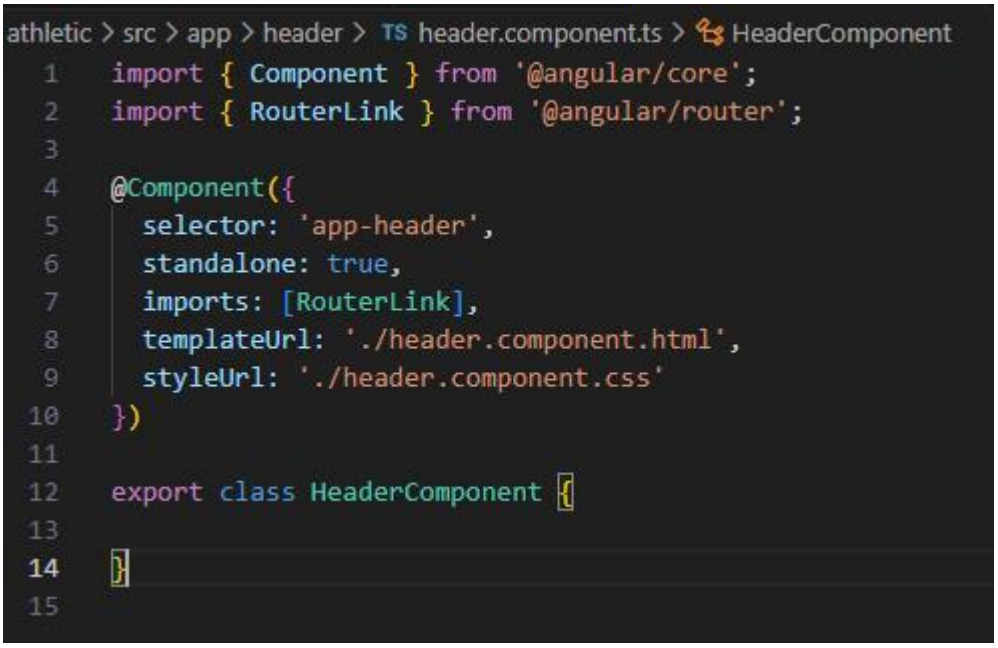
athletic > src > app > TS app.component.ts > ...

```
1   import { Component } from '@angular/core';
2   import { RouterOutlet } from '@angular/router';
3   import { HeaderComponent } from './header/header.component';
4   import { FooterComponent } from './footer/footer.component';
5
6
7   @Component({
8     selector: 'app-root',
9     standalone: true,
10    imports: [RouterOutlet, HeaderComponent, FooterComponent],
11    templateUrl: './app.component.html',
12    styleUrl: './app.component.css'
13  })
14
15  export class AppComponent {
16    title = 'athletic';
17  }
18
```

Importar RouterLink desde los componentes que vayan a llamar a otros para modificar el contenido de <router-outlet>

Los componentes desde los que queramos poder llamar cuando realicemos una acción, como **pulsar un botón,** deben **importar** en su correspondiente **.ts** el **RouterLink**.

En mi caso, la llamada se hace desde el componente header, pues es donde tengo el nav. Por lo tanto, en el archivo **header.component.ts** es donde importo RouterLink (líneas 2 y 7 de la siguiente captura).



Así es como se realiza la llamada al componente “about” desde **header.component.html**:

De forma similar a como utilizamos el “href”, añadimos la etiqueta [routerLink] y **le indicamos a qué componente va a llamar con el nombre que hayamos definido en app.routes.ts** (‘about’, en este caso).

app.component.html

Al ser la página principal, tendremos un header, un main y un footer, como de costumbre.

Los **componentes fijos**, en este caso el header y el footer, los llamamos abriendo y cerrando la etiqueta <app-header> y <app-footer>, respectivamente.

El **componente variable** en este caso se encuentra dentro del main y lo llamamos con la etiqueta de apertura y cierre <router-outlet>. Desde la web, al pulsar un botón que contenga un RouterLink, como el botón “Sobre Nosotros” del punto anterior, cargará el contenido del componente que ha sido llamado (‘about’, por ejemplo) en la parte de la web donde esté la etiqueta <router-outlet>.

Así es como se ve el app.component.html: