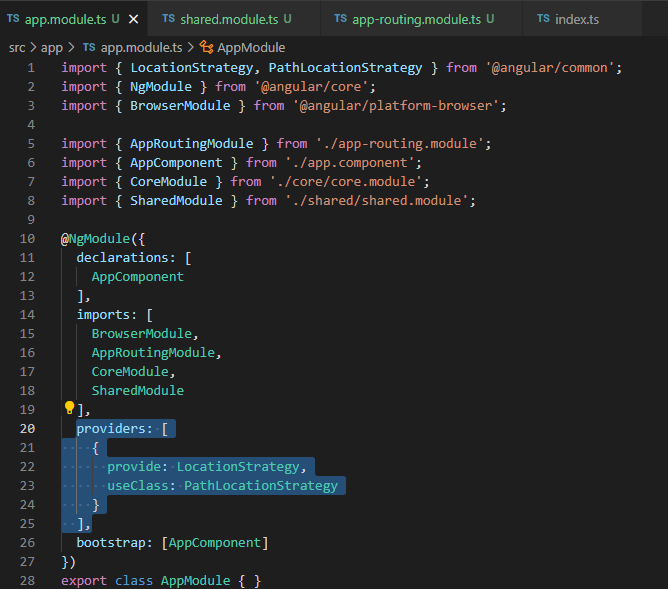
## Comandos

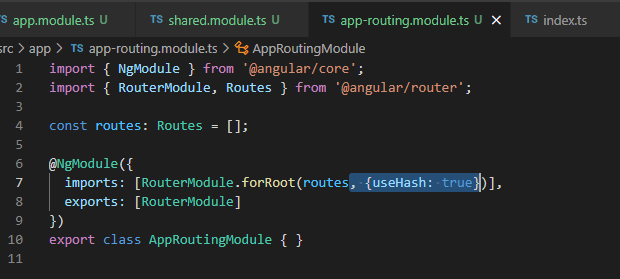
|  |  |  |
| --- | --- | --- |
| Comando | Descripción | Extra |
| ng generate module core | Posee los archivos core que siempre se tienen que cargar desde un inicio a la aplicación | luego de crear, agregar CoreModule a app.module.ts |
| ng generate module data | Crea el módulo de datos | generalmente se le agregan las carpetas de:  - constants (Constantes para utilizar dentro del código)  - interfaces (Models)  - schema (Donde estarán las Class del proyecto)  - services (Servicios para la información)  - api (Todos los servicios del api)  - mocks (Archivos para simular el backend de producción durante el desarrollo) |
| mkdir src/app/layout | Crear carpeta layout, la cual sirve para guardar componentes genéricos de la app | ej: header, sidebar, footer, navbar |
| mkdir src/app/modules | Crear carpeta modules, la cual posee todos los módulos de la aplicación |  |
| ng generate module shared | Crea el módulo de shared, el cual poseerá los módulos compartidos de la app.  Ej: [shared.module.ts](#_shared.module.ts)  Luego de crear, se debe agregar SharedModule a app.module.ts | ej: FormsModule, ReactiveFormsModule, HttpClientModule, etc.…  Generalmente se le agregan las carpetas de:  - components (Carpeta de componentes)  - services (Todo lo que se va a compartir en la aplicación a los diferentes módulos) |

## Código

### Quitar #

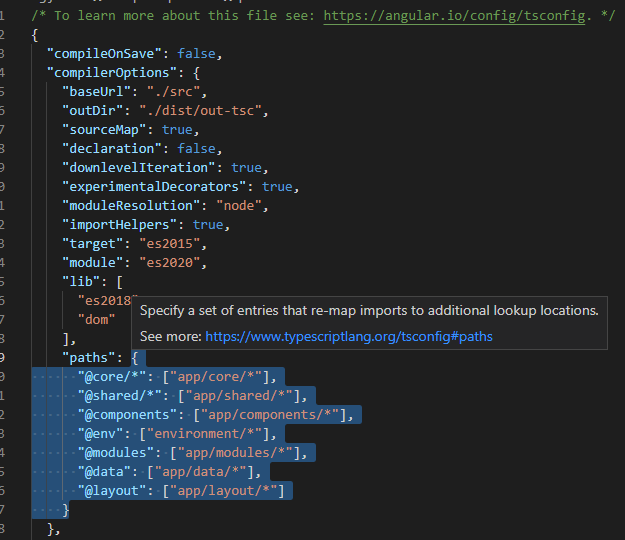
Para quitar el # de las rutas y utilizar la estrategia de pathlocation, se debe agregar



al módulo providers dentro de app.module.ts y luego en app-routing.module.ts se debe agregar la propiedad {useHash: true} a RouterModule.forRoot

### Alias PATH

Para evitar tener componentes demasiado largos (ej: ../../../../../../components) se utilizarán paths para acortar las rutas, para esto se debe agregar al archivo tsconfig.json el siguiente codigo.



Y luego cambiar la baseUrl de ./ a ./src



## Ejemplo de archivos

### shared.module.ts

