PEC 7
Frameworks: Routing en Angular

Desarrollo front-end con frameworks Javascript

Máster Universitario en Desarrollo de sitios y aplicaciones web

Semestre 20201

Estudios de Informática, Multimedia y Telecomunicación

TEORÍA

1. Introducción al Routing en Angular

En la anterior PEC construimos servicios que podían ser reutilizables y que nos permitían crear un código más organizado, reusable y sobre todo limpio (clean code). Además comenzamos a integrar nuestra aplicación frontend con el backend a través de peticiones HTTP haciendo uso del módulo `HttpClient`, y empezamos a gestionar la lógica de código asíncrono usando la biblioteca *RxJS*.

En este capítulo se va a presentar otra de las piezas fundamentales en el desarrollo de aplicaciones Web de lado del cliente, Routing, el cual nos va a permitir encapsular varias páginas y piezas en diferentes rutas, y permitirnos cambiar de una página a otra a través de enlaces sin dejar en ningún momento de estar creando una aplicación SPA (Single Page Application).

Aparte de renderizar el componente/pagina adecuada según la ruta a la que desee ir el usuario, el módulo de routing tiene algunas características extras. La primera es que nos permite “proteger” rutas en el lado del cliente, es decir, ciertas páginas no se renderizarán si no se satisface una determinada condición. Normalmente, las páginas se protegen para evitar que traten de acceder a ellas usuarios que no están autorizados, bien porque no están autenticados o porque su rol en la aplicación no se lo permite. No obstante, tenéis que tener claro que esto no es una medida de seguridad, sino evitar que un usuario acceda a una página que no le corresponde. Debemos tener en cuenta que la protección de información se debe realizar en el lado del servidor con las medidas de seguridad adecuadas, en el lado del cliente solamente estamos centrados en guiar adecuadamente a los usuarios. En este caso concreto, las guardas nos permitirán guiar a los usuarios a páginas en las que sí pueden estar.

Para la superación de este apartado de la práctica debes leer y realizar el paso a paso del código relativo al Capítulo 11 (https://learning.oreilly.com/library/view/angular-up-and/9781491999820/ch11.html).

2. Lazy Loading in Angular

Una de las técnicas para mejorar el rendimiento de las aplicaciones Web y de estructuración/organización de módulos es la carga perezosa o *Lazy Loading* de los módulos de Angular. Nuestras aplicaciones a medida que crecen van teniendo un gran número de pantallas que no todos los usuarios deben tener acceso, de hecho, nuestra aplicación Web va poco a poco creando módulos según los problemas que quieren resolver, clásicos ejemplo son los módulos de Users, Orders, Finances, etc. Está claro que no todos los usuarios de nuestra aplicación deben poder realizar una gestión de usuarios o de las peticiones de nuestros clientes. Por tanto, es buena idea organizar nuestra aplicación en módulos, los cuales se cargarán por demanda (cada usuario que lo requiera) en lugar de tener todos los módulos cargados a priori.

Los pasos que tradicionalmente se siguen para conseguir estructurar adecuadamente nuestra aplicación son los siguientes:

1. En lugar de definir todas las rutas en el punto principal, se debe pensar la aplicación en pequeños módulos y definir en cada uno de estos módulos sus propias rutas de modo que serían unidades autocontenidas e independientes de las otras.
2. Los componentes que utilizan cada módulo deben ser registrados en cada uno de los submódulos, en lugar de tenerlos en el módulo general.
3. Se deben registrar todas las rutas de las rutas hijas (rutas de cada submódulo).
4. A nivel de aplicación, en lugar de invocar a una ruta concreta de un submódulo debemos indicar que se debe cargar un submódulo, y éste será el encargado de cargar las rutas de su módulo.

Para la superación de este apartado de la práctica debes leer y realizar el paso a paso del código relativo al Capítulo 12 (https://learning.oreilly.com/library/view/angular-up-and/9781491999820/ch12.html) que trata concretamente sobre Lazy Loading.

La sintaxis de uso de Lazy Loading ha sido modificada en la versión 8 de Angular y, por tanto, la mayoría de los recursos se encuentran desactualizados. No obstante, el cambio no es grave, ni tedioso pero sí es

importante aplicar el cambio que se debe realizar. En la siguiente dirección se encuentra explicado el cambio que se debe realizar, así como un vídeo al final del artículo con un paso a paso de los cambios.

Lazy-Loading nueva sintaxis Angular >8

https://dev.to/nishugoel/lazy-loading-in-angular-8-5g54

NOTA: Como miembros de la Comunidad UOC, tenéis disponible un gran catálogo de recursos de la editorial O'Reilly. Para acceder debéis seguir los siguientes pasos:

* Acceder a https://learning.oreilly.com

* Cuando te pida el *login* y *password*, escribir en el *campo Email Address or Username* vuestro correo electrónico de la UOC, p.ej. dgarciaso@uoc.edu

* Después pulsáis fuera con el ratón y desaparecerá el campo *password*. Además el botón de *Sign In* cambiará por un botón rojo con el texto *Sign In with Single Sign On*.

* A partir de aquí seguid los pasos que os vaya pidiendo, seguramente os llevará a una página de autenticación con el logo de la UOC. Además os llegará un e-mail a vuestra cuenta de correo electrónico de la UOC avisando de que os habéis suscrito a este catálogo.

* Una vez tengáis acceso al catálogo de O'Reilly, para ver el contenido de los enlaces que os proporcionamos, sólo tenéis que estar autenticados en O'Reilly y abrir una pestaña nueva en vuestro navegador y copiar el enlace que os facilitamos.