

ReactJS Developer

Módulo 4


History Fallback

Fallback

Cuando utilizamos el `BrowserRouter` (y con ello la *API History*) ocurre algo particular.

Si navegamos en la aplicación y observamos la *URL* veremos que parece ser **una *URL* completamente válida y limpia**. Eso da una experiencia de usuario muy cómoda.

Ahora bien, si publicamos nuestro proyecto e intentamos acceder directamente a esa *URL* nos tiraría **error** (inténtalo) ¿Por qué?



Al acceder directamente a una *URL* como las generadas por el `BrowserRouter`, que no se diferencian en nada de una *URL* común, el navegador **solicita ese recurso**, que **no existe realmente**, y por eso tenemos **el error**.

Si decidimos utilizar el `BrowserRouter` necesitamos diseñar un servidor (un back-end) para nuestra aplicación React capaz de enviar el archivo *index.html* **antes de mandar error 404**.

Es decir, necesitamos un ***fallback***.

Esto es programación del lado del servidor y puede variar de lenguaje en lenguaje. Aquí te ofrecemos un algoritmo genérico del lado del servidor para implementar el *Fallback*:

- El *fallback* puede ser implementado como un **middleware** o como una **ruta que se evalúe cuando no hubo match con otras rutas**.
- Básicamente, el código central del *fallback* consiste en **responder con el archivo *index.html***.
- Esta respuesta debe enviarse **antes de lanzar error 404**, es decir, **cuando no hubo match con ninguna otra ruta**.

Cabe aclarar que éste es un algoritmo **genérico** y su implementación varía **según la tecnología usada para el back-end**.

Por mencionar solo algunos ejemplos:

- En *ASP.Net Core* tienes un servicio específico para las *SPA* que contempla justamente esto de forma automática. No necesitas programar nada. Ya viene con el *framework*.
- En *Node.js* nativo necesitas definir una función con *http.createServer* para contemplar todos los casos posibles.
- En *Express.js* puedes crear un middleware, o poner una ruta al final de todas las demás ¡Mira [este](#) paquete *NPM*!
- En *PHP* puedes configurarlo directamente mediante el archivo *.htaccess* de *Apache*.



**¡Sigamos
trabajando!**