

Laboratorios

Ejercicio 01

Para este laboratorio vamos a usar la página <https://jsonplaceholder.typicode.com/> que nos va a permitir simular el backend de la aplicación.

- Primero, crear un componente que haga una llamada al backend, siempre usando la librería HttpClient de Angular, para obtener los datos de un post y mostrarlo en el template. (<https://jsonplaceholder.typicode.com/posts/1>)
- Como no es conveniente hacer las llamadas al backend desde un componente, vamos a crear un servicio que se encargue de esas tareas.
 - En ese servicio vamos a definir los siguientes métodos:
 - `getPost`: recibirá un id y obtendrá los datos del post que correspondan a ese id ('<https://jsonplaceholder.typicode.com/posts/>' + `idPost`)
 - `getPostResponse`: también recibirá un id y va a obtener los datos del post con ese id, pero en este caso lo va a hacer convirtiendo el objeto recibido en un tipo de la interfaz `IPost`.
 - `getPostError`: este método deberá intentar obtener datos de una url inexistente. El error debe ser manejado por un método callback que también definiremos en el servicio.
 - Por cada uno de estos métodos crear un componente que muestre los datos obtenidos o el mensaje de error en su defecto.

Ejercicio 02

Vamos a seguir utilizando la página <https://jsonplaceholder.typicode.com/> para simular el backend de la aplicación.

- Crear un servicio que tenga los siguientes métodos:
 - `addPost`: recibirá un objeto del tipo `IPost` como parámetro y a través del método `post` de la librería HttpClient deberá enviarlo al servidor (<https://jsonplaceholder.typicode.com/posts>). No olvidar que para el envío de datos es esencial definir los headers de la petición. Para ello debemos almacenar en una variable o constante una nueva instancia de `HttpHeaders` con las propiedades necesarias.
 - `updatePost`: recibirá un objeto del tipo `IPost` como parámetro y a través del método `put` de la librería HttpClient deberemos enviarlo para que actualice un registro existente.
 - `deletePost`: recibirá un id como parámetro y a través del método `delete` de la librería HttpClient eliminará el registro con el id ingresado.
- Por cada uno de estos métodos crear un componente que muestre los datos obtenidos o el mensaje de error en su defecto.