

Ejercicios 6

En esta última sección vamos a modificar la aplicación de iVideoclub para que descargue el listado de películas utilizando una API de Internet.

Ejercicio 6.1 - Nuevo servicio

Para descargar los datos de la API de Internet vamos a utilizar la clase `HttpClient` desde dentro de un nuevo servicio. Empezaremos creando este nuevo servicio, al que llamaremos "películasAPI", y añadiéndole la siguiente variable:

```
1 APIep = 'http://gbrain.dlsi.ua.es/videoclub/api/v1/catalog';
```

Este servicio tendrá los mismos dos métodos que el servicio `PeliculasService` que ya teníamos: `getPelículas()` y `getPelicula(id)`, los cuales además devolverán el mismo tipo de dato (`any`). Sin embargo, para obtener los datos en este caso accederemos a Internet mediante la librería `HttpClient`.

Para utilizar esta librería tenemos que importarla e inyectarla en el constructor. Una vez hecho esto ya podremos realizar la petición al *endpoint* (la variable `APIep` que hemos definido antes). Empezaremos modificando el método `getPelículas()` para que realice una petición tipo `get` a esta dirección y devuelva el resultado:

```
1 return this.http.get(this.APIep);
```

En el método `getPelicula(id)` tendremos que realizar la misma petición pero añadiendo al *endpoint* el `id` recibido por parámetro. Esto lo podemos hacer simplemente concatenando:
`this.APIep + '/' + id`

Ejercicio 6.2 - Página videoclub

Una vez definido el servicio vamos a modificar la página `videoclub` para que lo utilice. Para esto en primer lugar tenemos que eliminar el import y la inyección de dependencia del servicio anterior (`PeliculasService`). A continuación importamos el nuevo servicio y lo inyectamos en el constructor.

Una vez hecho esto modificaremos el código que teníamos en el constructor para recuperar la lista de películas. El método "getPelículas" del nuevo servicio ya no devuelve directamente la lista de películas, sino que devuelve un objeto *Observable* con el resultado de la petición asíncrona. Este tipo de objetos tienen un método `subscribe` que nos permite "suscribirnos" a la petición y realizar una acción cuando se reciba el resultado. En nuestro caso lo que haremos será asignar directamente el resultado a la variable `listaPelículas` que ya teníamos definida. Además mostraremos cualquier posible error por el log de consola:

```
1  películasAPIService.getPelículas().subscribe(
2      result => {
3          this.listaPelículas = result;
4      },
5      err => {
6          console.log(err);
7      }
8  );
```

Por último tenemos que realizar un par de modificaciones en la vista asociada. La primera modificación es corregir la ruta de las imágenes, ya que en la versión actual se añade de forma manual la ruta de la carpeta "assets/posters":

```
1  <ion-img src="assets/posters/{{película.poster}}"></ion-img>
```

Por lo que tendremos que quitar "assets/posters/" para que utilice directamente la URL completa suministrada por la API.

El segundo cambio es sobre el bucle que genera el listado de películas. Actualmente tenemos un bucle en la etiqueta `<ion-item>` como el siguiente:

```
1  <ion-item *ngFor="let película of listaPelículas; index as id"
    (click)="verPaginaDetalle(id)" detail>
```

Con el nuevo servicio **no** podemos utilizar el índice del array, sino que tendremos que usar el campo `id` que contiene el nuevo listado de películas obtenido de Internet. Es decir, tendremos que quitar "index as id" y, en lugar de usar `id` en la llamada al método `verPaginaDetalle()`, tendremos que usar `película.id`:

```
1  <ion-item *ngFor="let película of listaPelículas"
    (click)="verPaginaDetalle(película.id)" detail>
```

Este mismo cambio lo tendremos que realizar también para la vista alternativa con el modo cuadrícula.

Ejercicio 6.3 - Página detalle

Por último vamos a actualizar la página detalle para que utilice también el nuevo servicio. Igual que en el ejercicio anterior, elimina el import y la inyección de dependencia del servicio `PeliculasService` y a continuación importa el nuevo servicio e inyéctalo en el constructor.

Una vez hecho esto vamos a modificar el código que teníamos dentro del método "`ngOnInit`" para cambiar el uso del método `getPelicula(id)`. Como ya vimos, los métodos del nuevo servicio no devuelven directamente un resultado sino que devuelven un objeto tipo *Observable*. Por lo que utilizaremos el `subscribe` para escuchar a esta petición y asignar el resultado cuando se obtenga la respuesta.

Por último tenemos que realizar también un par de modificaciones en la vista. La primera modificación es corregir la ruta de las imágenes para que utilice directamente la URL completa suministrada por la API. Y el segundo cambio tiene que ver con el binding del objeto `pelicula`. Actualmente la vista muestra directamente el contenido de este objeto, por ejemplo, en el título tenemos:

```
1 <ion-title>{{pelicula.title}}</ion-title>
```

Sin embargo, cuando se carga la vista este objeto no estará definido ni contendrá ningún valor dado que la descarga de los datos desde la API de Internet tardará un poco. Para evitar que nos de un error tenemos que añadir el símbolo `?` al hacer el *binding* (`{{pelicula?.title}}`). De forma que inicialmente, cuando la variable todavía esté vacía, no muestre ningún valor, pero que cuando se descarguen estos datos se actualice la vista y se muestre.