**Aplicación Noticias Ionic**

Los temas puntuales son:

1. Crear una aplicación real
2. Consumir APIS para desplegar información
3. Uso de componentes personalizados y módulos
4. Uso de componentes de ionic
5. Uso de plugins
6. Guardar en el storage del dispositivo nativo
7. Generar un diseño que funcione en tabletas y escritorio también
8. Uso de Toasts
9. Mostrar mensajes cuando no hay información guardada
10. Uso del infinite-scroll para obtener más información cuando llegamos al final de cada página de noticias
11. Uso de segmentos con scroll

Creamos el nuevo proyecto

Ionic start

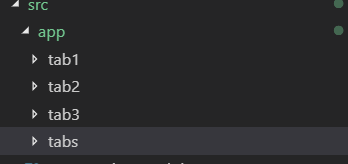


Para obtener las noticias del api debemos obtener el apikey

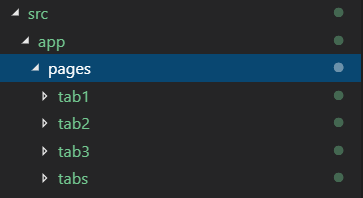
<https://newsapi.org/>

Luego de registrarse nos genera el apikey.

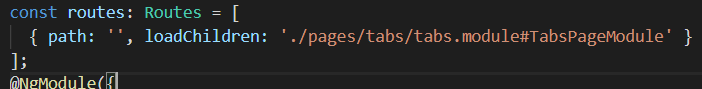
Al crear la aplicación con tabs, este crea las carpetas dentro de la ruta app. Es necesario moverla a la carpeta pages (que crearemos).



Resultando

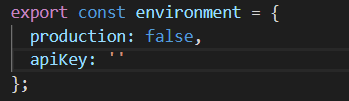


Ahora dentro del app.routing.module hay que cambiar la ruta del modulo de tabs

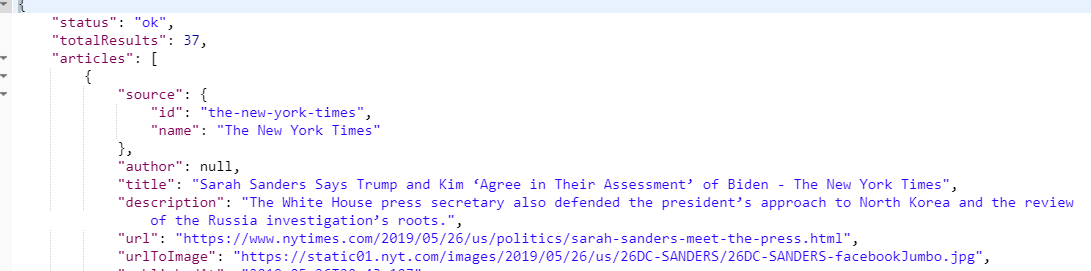


Servicio para consumir las noticias de newsapi.org

En la carpeta environments el archivo environment.ts creamos la propiedad apKey y pegamos la que obtuvimos al registarnos . Tambien en el prod.



Si utilizamos postman podemos utilizar el url que vemos en la pagina de newsapi.org



Entonces creamos nuestro servicio de noticias.



Para utilizar servicios hay que importar desde el app.module el HttpClientModule.

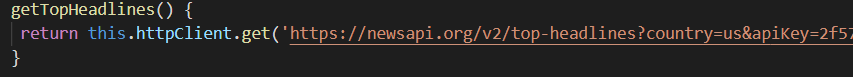


y agregarlo a los imports

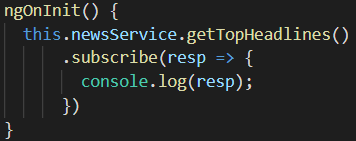
En el servicio importamos el HttpClient para luego usarlo en un método.



Y el servicio



Probando el servicio lo utilizaremos en un tab.

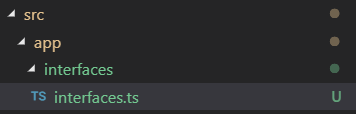


De devolver un error correr nuevamente el ionic serve.

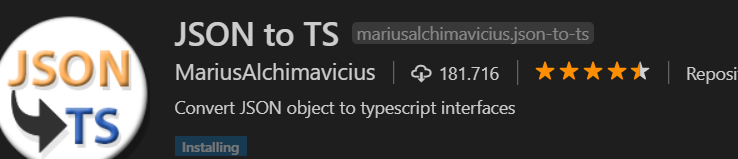
Interfaces para el manejo de la información

Es necesario que la información que va a regresar del servicio tenerla mapeada para tener los datos tipificados.

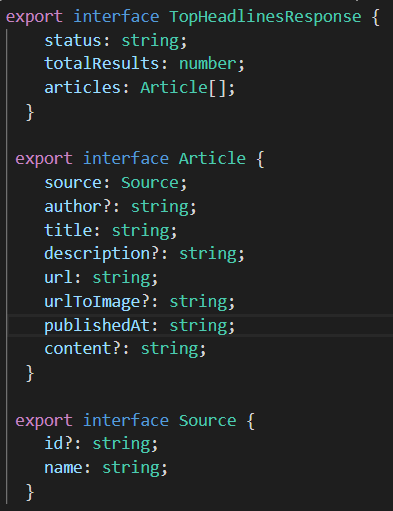
Creamos entonces un archivo interfaces dentro de la estructura del proyecto.



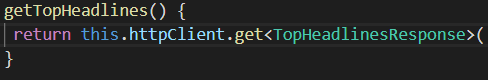
Como ya probamos la respuesta del servicio, tenemos el json que devuelve, podemos crear la interfaz a partir de esta respuesta con el plugin:



El cual al presionar control + shift + p , y escribir json to … podemos convertir lo que tenemos seleccionado a su interfaz. Cambiamos el nombre del objeto principal por el nombre en este caso del endpoint por la respuesta que estamos recibiendo del servicio.

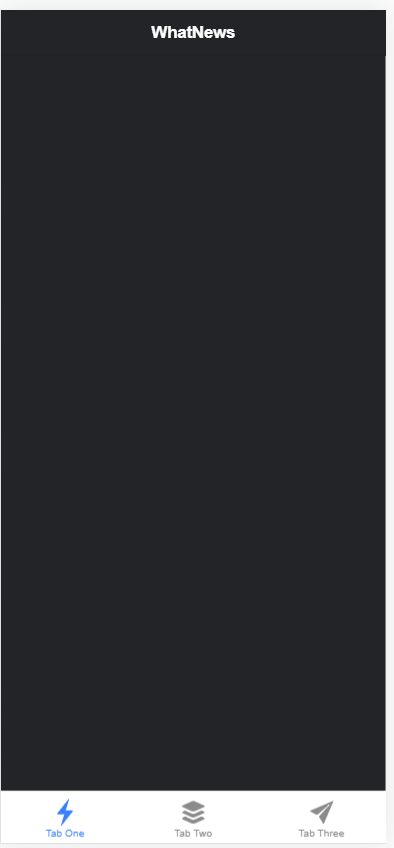
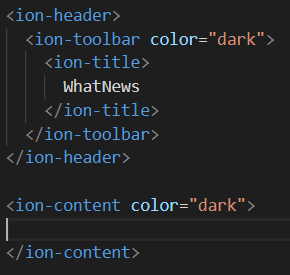


Ahora ya que tenemos la interfaz podemos tipificar la respuesta del servicio.



Por lo que al momento de obtener su repsuesta en el llamado del tab 1, podemos obtener ayuda del editor al navegar por el objeto de respuesta,

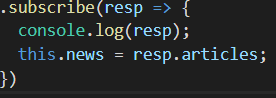
Mostrando los artículos de noticias en el html



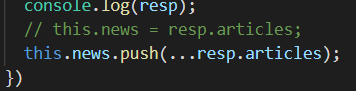
Como esta aplicación se puede mostrar por web, crearemos una grilla que pueda ser vista bien en todas las plataformas.

Creamos una arreglo de tipo Article el cual obtendrá el valor que responde el servicio del arreglo de noticias.

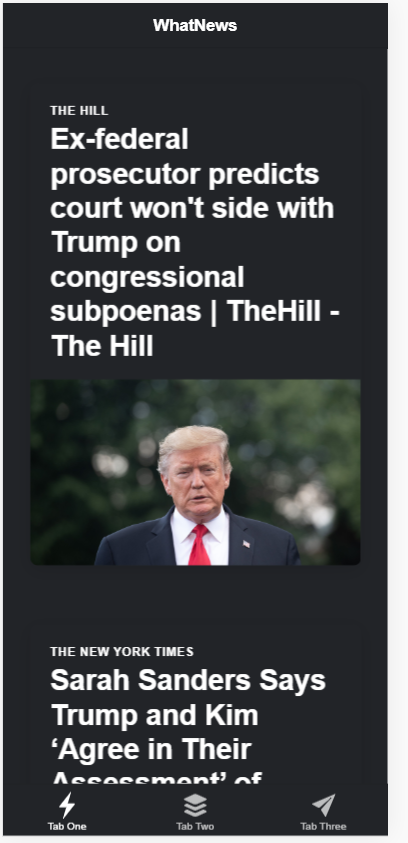
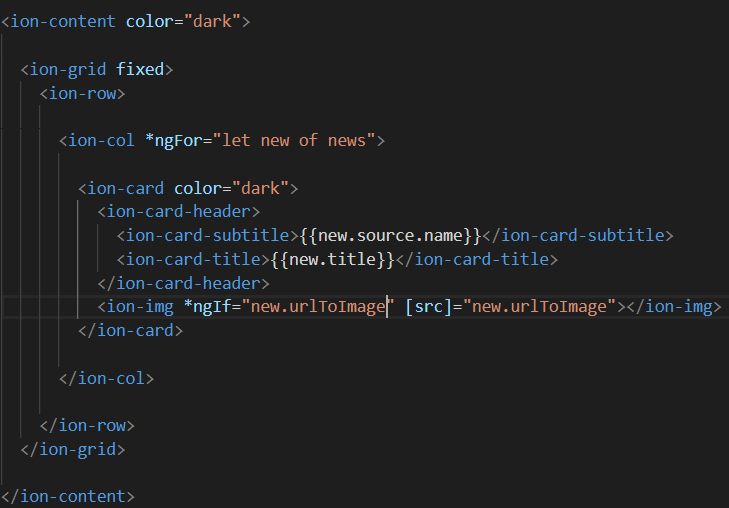




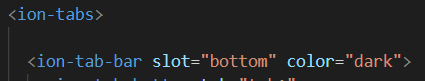
Si se asigna cada vez que responda el servicio, se estará sobreescribiendo el arreglo, para esto le haremos un push de la respuesta.



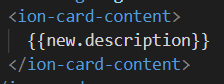
Y Para mostrar en el html



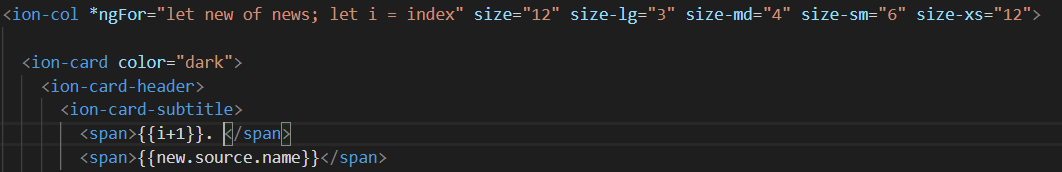
Arreglando el tab, desde la carpeta tab modificamos el ion-tab-bar.



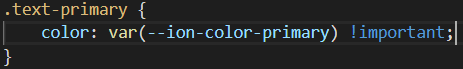
Agregamos la descripción y la especificación del tamaño del ion col.


Agregando el índice y numerando.



Podemos crear una variable global para añadir estilos mediante el global.scss



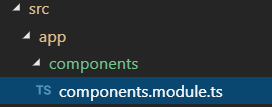
Al elemento que se agrege el class text-primary obtendrá este color (azul).



Componente de Noticias y de noticia

Para no tener código duplicado crearemos un modulo que maneje los componentes de las noticias.



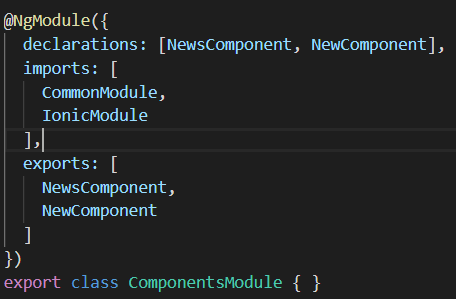


Ahora creamos el primer componente

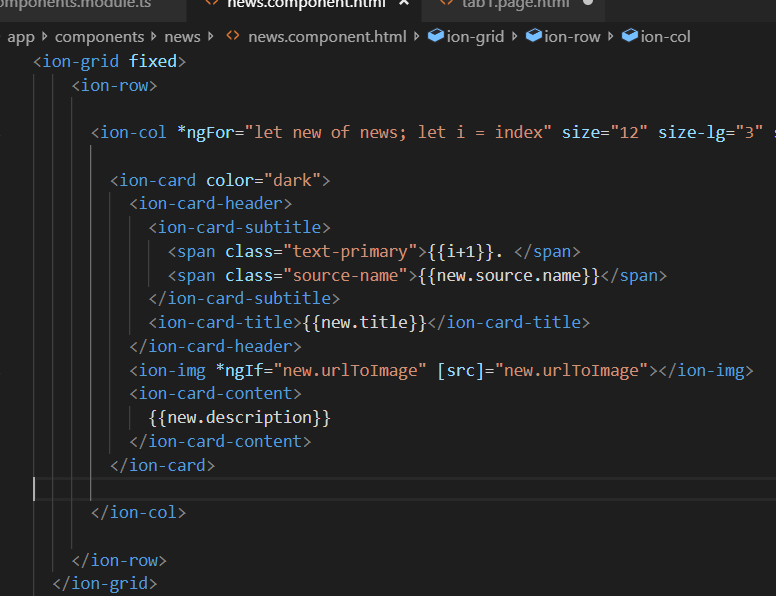


Y hacemos igual para crear new, el componente de la noticia individual.

Cada componente que se agrega dentro de components hay que declararlo y exportarlo en el archivo components.module. Es importante también agregar el IonicModule ya que se estarán usando sentencias de ionic dentro del html. Revisando luego El NewComponent no es necesario exportarlo ya que este estará incluido en el NewsComponent.



Ahora el código que esta en tab 1 desde el ion-grid, lo cortamos y pegamos en el news.html



Ahora en el news component declaramos una variable input que recibirá el arreglo de noticias



Ahora para poder utilizar el componente news que esta dentro del modulo components, hay que importar el modulo components en el tab1.module en los imports.

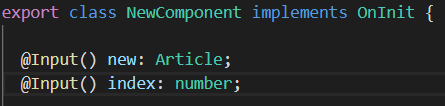
Si no carga es probablemente por los modulos, reiniciando el serve se arregla.

Ahora haciendo lo mismo con el card de las noticias para que este componentizado.

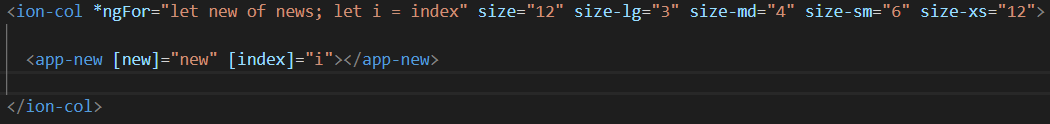
Dentro del new.html



Y dentro del new.ts recibiendo como input la noticia a mostrar y el índice de esta.

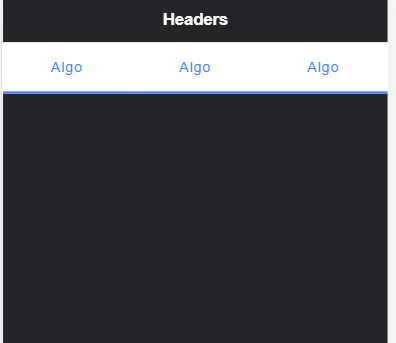


Y mostrando la desde el news html.

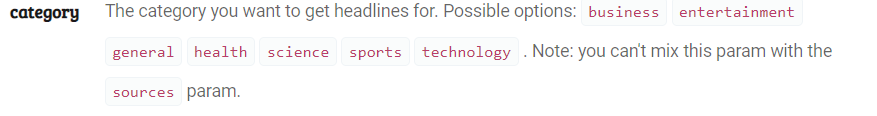


Tab de Encabezados

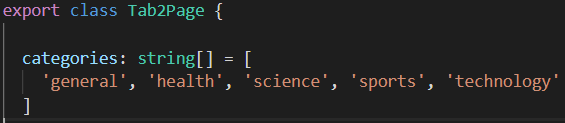
En el html del tab 2 creamos el segment y dentro los botones que serán de encabezado, en este momento como prueba para ver como queda con varios y luego traerlos desde un servicio.

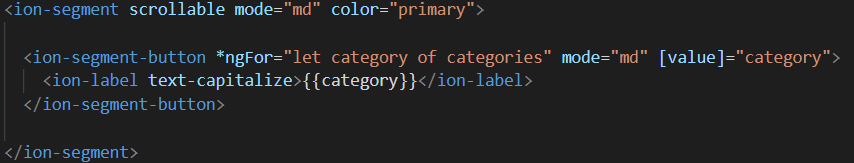


En el api nos muestran la información del parámetro que podemos pasar para pedir sobre una categoría.

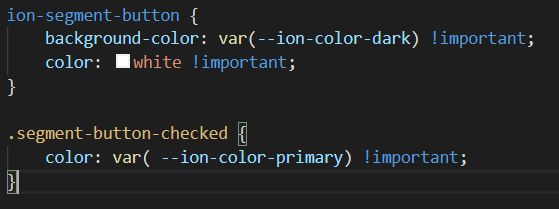


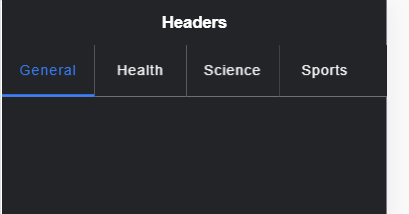
Creamos un arreglo con estos valores.



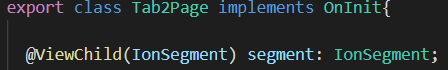


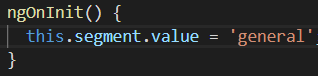
Agregando estas clases al css del archivo logramos cambiarle el color, las líneas blancas laterales solo se muestran en Chrome.





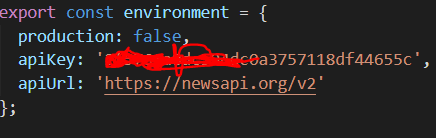
Para seleccionar por default al iniciar la pagina



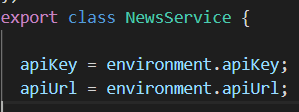


Ya que vamos a estar consultando el mismo url utilizaremos las variables,

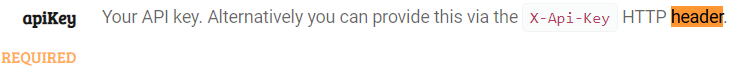
Creamos en el environment la variable apiUrl



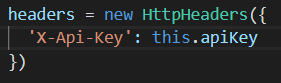
Y desde nuestro servicio las asignamos.



En la documentación del api nos dice que podemos pasar el key como header.

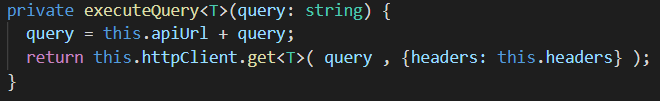


Asi que creamos una variable header de tipo httpheaders.

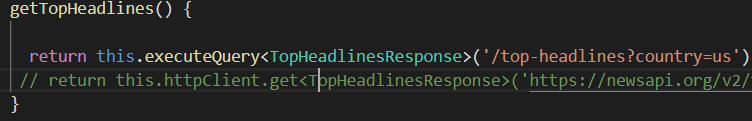


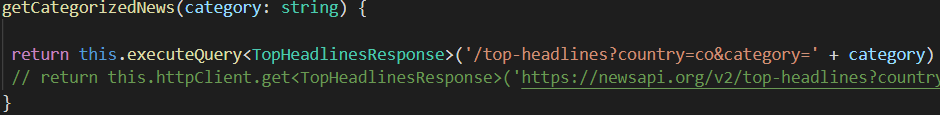
Ahora crearemos una función que ejecute los query, recibiendo como parámetro el nombre del endpoint a llamar.   
En estafuncion la tipamos de tipo T, es decir que va a recibir su tipado en la respuesta. Y luego como parámetro de entrada reibe el query que le envia la función llamada y que especifica el endpoint.

Al hacer el get mandamos el query y como segundo parámetro el header que contiene el apikey.

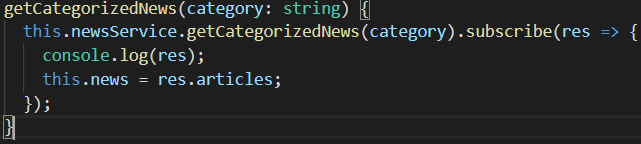


Luego modificamos las funciones que teníamos para que llamen a esta función de executequery.  
Ejecutandola y pasandole como parámetro el endpoint.





Para categorizar las noticias creamos una función que envie la categoría seleccionada y traiga las noticias.

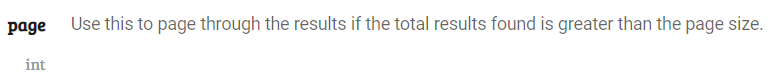


Y en el html

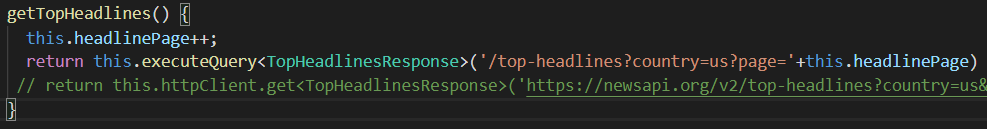


Implementando el infinite scroll Entendiendo el paginado.

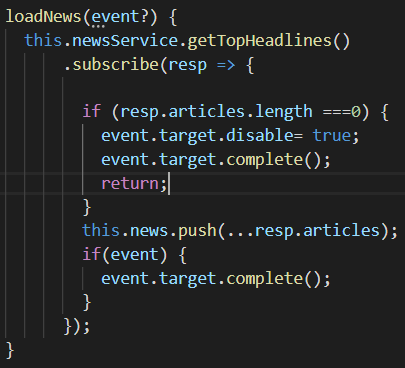
El servicio puede recibir un parámetro page que es el que controla la pagina que está mostrando de resultados.



Desde el servicio podemos controlar el paginado creando una variable local que se inicializa en 0. Y al llamar a la función de cargar noticias va aumentando



Y ahora en el ts del tab 1,   
Si la respuesta del servicio es que ya no tiene mas artículos, completa el infinite scroll y lo desactiva,   
Si es la primera vez que carga ósea la función es llamada sin el event como parámetro solo se ejecuta y sube el contador del paginado en el servicio.  
Si el la función es llamada con el evento y devuelve que aun hay articulos solamente completa el infinite scroll.



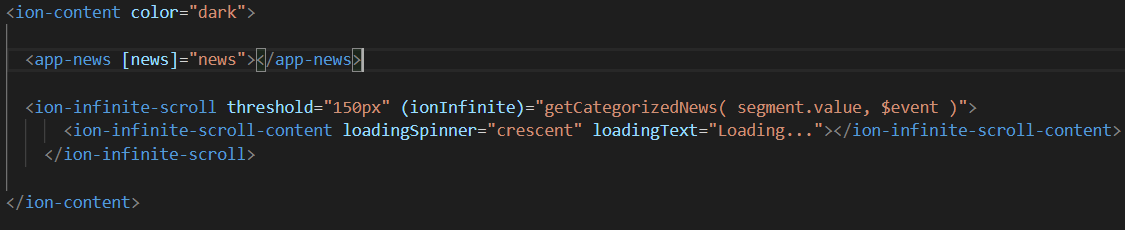
Investigar la forma de capturar el error al recibir del servicio:



Infinite scroll de las categorías

Importante al cambiar de categoría hay que hacer un refactor ya que se puede hacer push de las noticias solo validando cuando se **cambia** el segment, poner el arreglo en **cero**, de lo contrario hacer el **push** para cargar las noticias restantes.

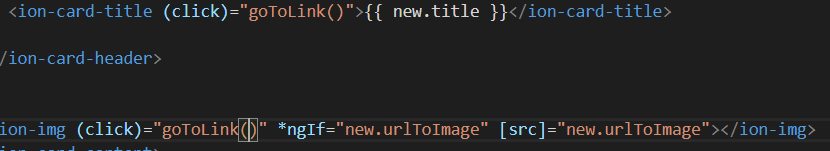






Abrir el sitio web de la noticia en el navegador, al hacer click en la noticia.

En el objeto respuesta de cada noticia que recibimos tenemos un url.  
Capturando el evento click desde noticia.



Para poder abrir el navegador desde el dispositivo es necesario instalar el plugin de in app broser, desde la pagina de ionic

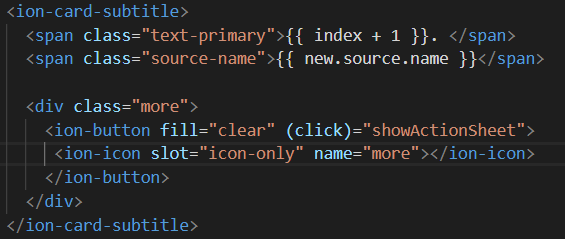


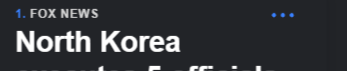
La primera línea permite utilizar el plugin con ionic, y la segunda es para typescript.

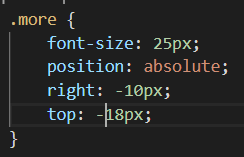
**El primer paso para utilizar un plugin nativo es declararlo en el app.module, desde el import path que dice la documentación.**



**Mostrando un action sheet**



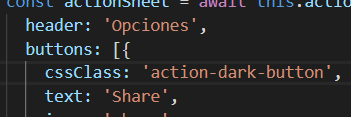


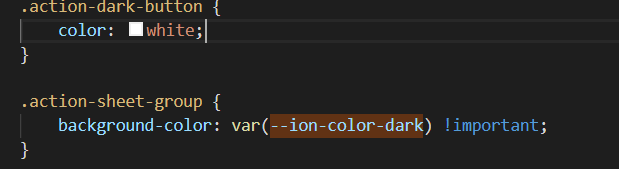


Luego podemos implementar el action sheet



Si le agregamos la propiedad css class podemos darle estilo al action sheet. Para detectar estos estilos hay que declararlo de forma global en el archivo global.scss ya que local al componente no lo detectará.



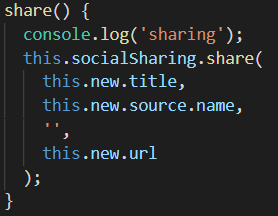


Compartir en redes sociales

Con el plugin social sharing instalado desde ionic.



Recordar importarlo en el app.module en los improts.





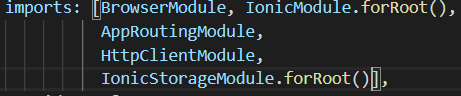
Guardar noticias en el dispositivo (Storage)

Para cuando grabe en el storage se grabe en una base de datos local





Y luego agregarlo al app.module en los imports.



Trabaja en base a promesas asi que se puede trabajar con async.

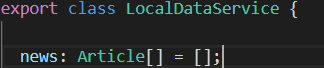
Se puede manejar el storage desde el mismo servicio pero no es lo conveniente, ya que el servicio que tenemos es para cargar datos. Asi que creamos otro servicio.



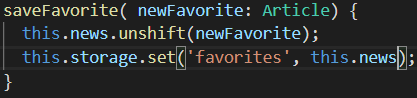
Ya dentro de nuestro servicio improtamos

import { Storage } from '@ionic/storage';

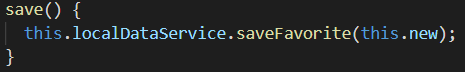
Creamos un arreglo vacio de noticias



Creamos un método de guardar favoritos, en donde recibimos una noticia de tipo Article y la añadimos al principio del arreglo de noticias que creamos. Y Luego la guardamos en el storages con la siguiente estructura.

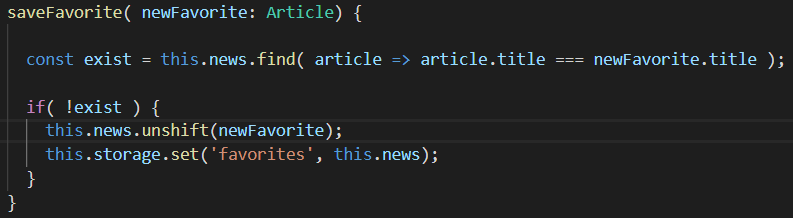


Luego en nuestro componente de noticia (new), agregamos el servicio y lo llamamos al momento de hacer click en la opción de action sheet.



En este momento permitimos guardar la misma noticia mas de una vez, hay que añadir una validación.

Utilizando la función find del arreglo buscamos y de no encontrar podemos guardarla.

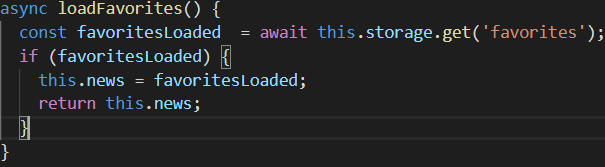


Cargando las noticias desde el storage.

Desde el tab 3 trabajaremos las noticias cargadas desde el storage.



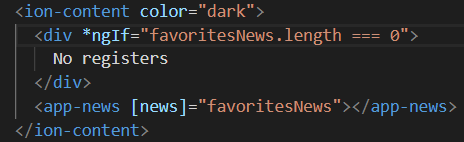
En el servicio hacemos que la función sea asíncrona, y luego declaramos una variable que espere la respuesta del storage.  
Validamos que esta tenga valor al momento de cargar, por que en el caso que se elimine la bd, esta devolverá null y hará que nuestro arreglo news sea null.



Finalmente cargamos los datos del servicio en nuestro arreglo local al tab3.

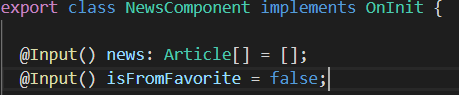


Tambien mostraremos un mensaje cuando no hay ningún registro



Modificando el action sheet de favoritos para podes desmarcar favorito

En el componente de news, agregamos otro parámetro recibido por input y lo inicializamos en falso, para que asi no afecte a las otras pantallas.



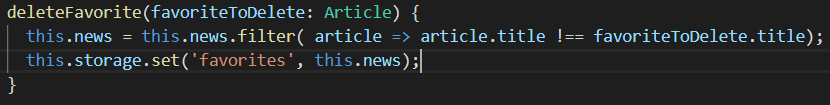
Al igual que el de new pero sin inicializar el valor ya que siempre recibirá un valor.

Luego en los html desde el tab 3 y news pasar el valor.





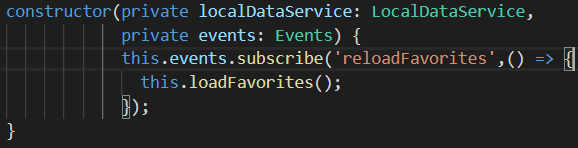
Desde el servicio cremos la función de borrar una noticia. Se hace un filtro del arreglo que devuelva el arreglo sin esa noticia y luego se guarda en el storage.



Para borrar el articulo sin cargar nuevamente la pagina hay que emitir un evento que carge nuevamente las noticias.

Se emite el evento desde el new.component y se recibe desde el tab 3





Algunos detalles esteticos:

Para remarcar el color del tab seleccionado.  
Por defecto ionic le asigna la clase tab-selected.  
Podemos desde el global.scss asignarle estilo