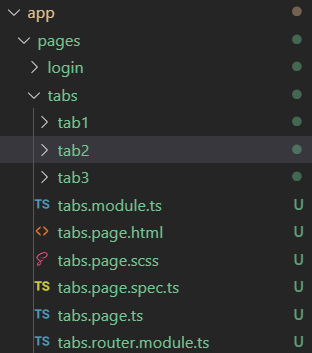
**Front de Fotosgram**

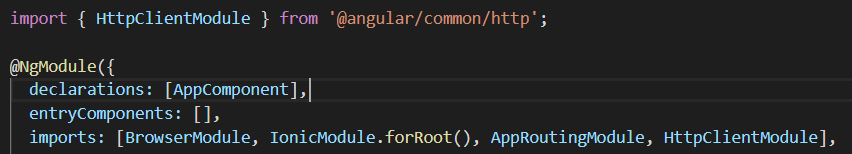
Ionic start fotosgram-front tabs

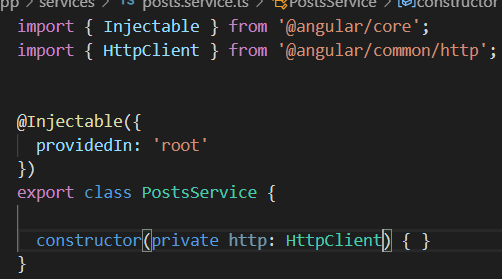
Ionic g p pages/login  
Luego reestructuramos el proyecto de la siguiente forma

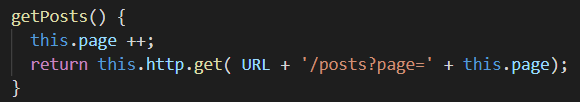


Creamos el servicio  
ionic g s services/posts

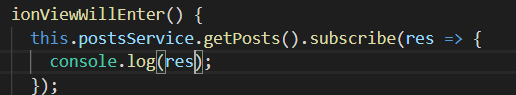
En el app module importamos



Y en el servicio  




En el componente de tabs solo como prueba consultamos el servicio, el cual nos mandara una respuesta de error por el cross origin.





<https://chrome.google.com/webstore/detail/allow-control-allow-origi/nlfbmbojpeacfghkpbjhddihlkkiljbi/related>  
Con ese plugin se puede verificar

Para resolver el problema del cors es necesario modificar el backend.

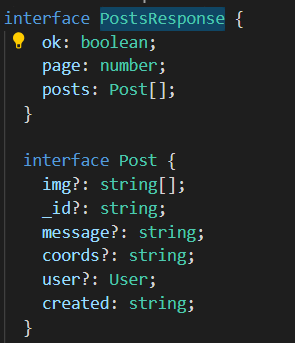
En el index.



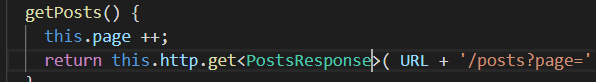


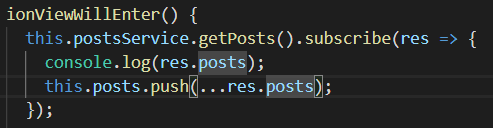
Y luego correr nuevamente el node dist  
Correr en Firefox y todo bien

Ahora desde postman le pegamos al servicio y creamos una interfaz con la ayuda del visual code

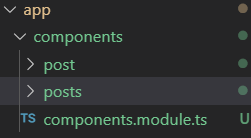


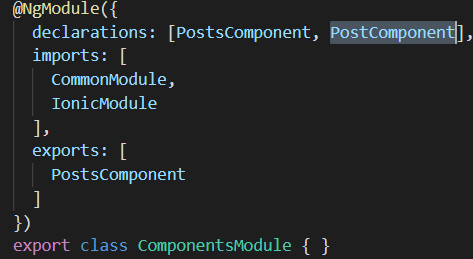
Y en el servicio tipamos la respuesta.



En el tab1 creamos el arreglo de posts y lo llenamos con la respuesta del serivcio, utilizando el push y spread para que se   


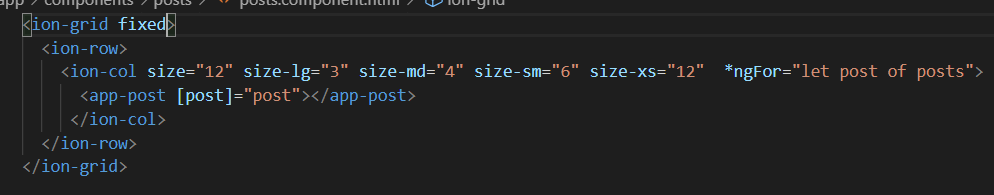
Vamos a crear un componente post y posts y un modulo de componentes

  
Siendo el modulo

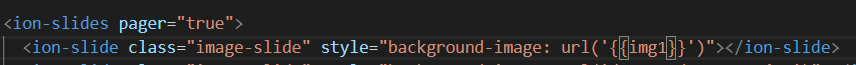


Importamos el components module en el modulo de tab 1 y luego ya podemos usar el tag en el hmtl.

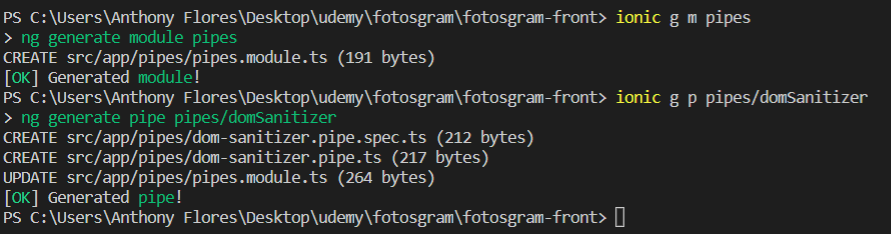
Estructurando para diferente pantallas los posts.



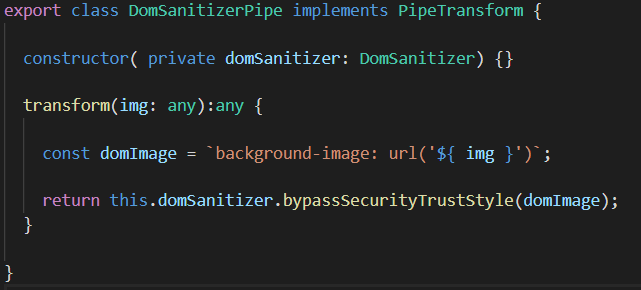
**Pipe para imágenes DomSanitizer**

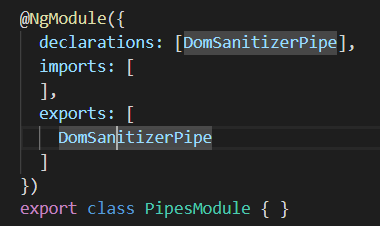


Ahora para procesarlas imagenesen una variable de typescript necesitamos pasarla por un pipe, asi que generamos el modulo de pipes y creamos el pipe sanitizer



En el que se recibe la imagen y se concatena con el path, y luego pasamos por el sanitizer





Y en el components module importamos el modulo de pipes para poder utilizarlo

Infinite scroll

