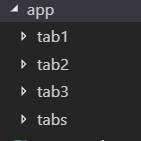
Scanner



Como al momento de crear los tabs los crea directamente en src, creamos manualmente la carpeta pages y metemos ahí los tabs

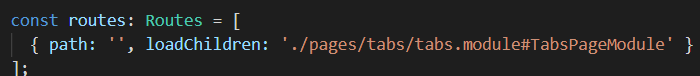


Luego hay que modificar el app routing con la nueva ruta

En el tabs html



Modificamos en el app routing el path de tabs module



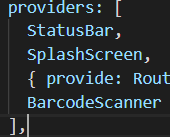
Utilizando el plugin

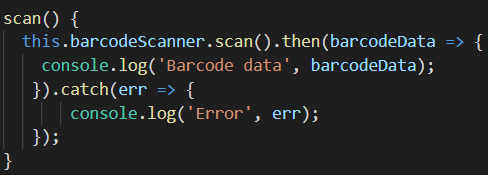
<https://ionicframework.com/docs/native/in-app-browser#installation>

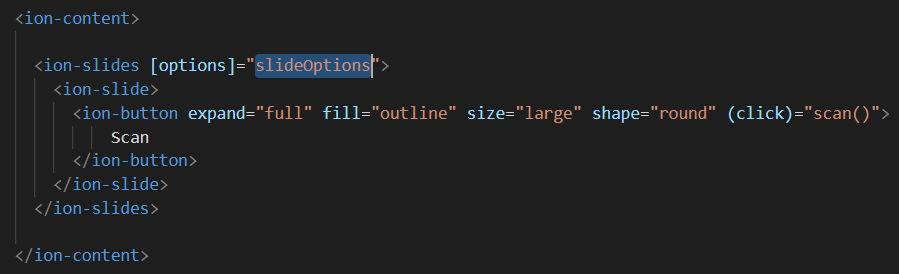
El barcode scanner lee Codigos Qr y codigos de barra



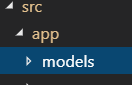
Y luego hay que implementarlo en el app module con los providers

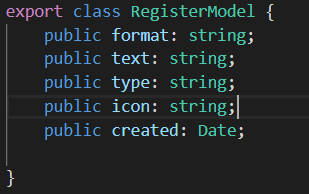




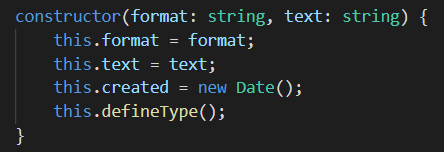


Creamos una carpeta models





Y luego el constructor

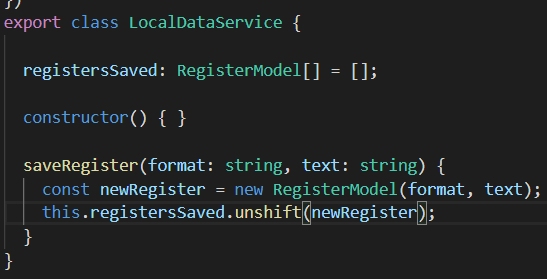


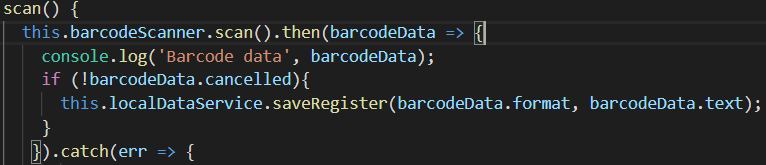
Luego en el método que detecta el tipo de qr definiremos con un switch



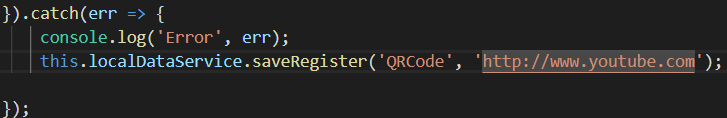
Ahora crearemos un servicio para almacenar los datos escaneados



Creamos el arreglo de tipo register model que almacenará los registros.  
En el método de guardar registros recibimos como parámetro los datos que lee el scanner y lo creamos como un nuevo registro . Luego lo almacenamos en el arreglo en la primera posición con el unshift.  


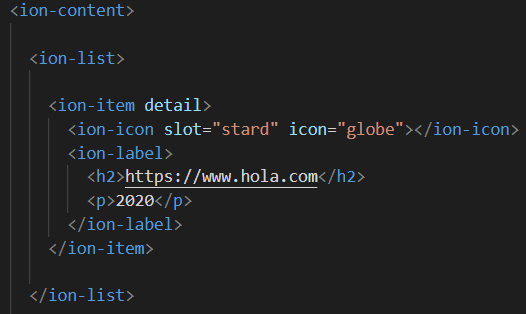
Y desde el tab llamamos al servicio  


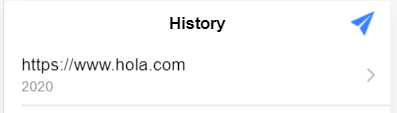
Para efectos de prueba en el error para ver desde la pc:



Mostrar registros en el historial:

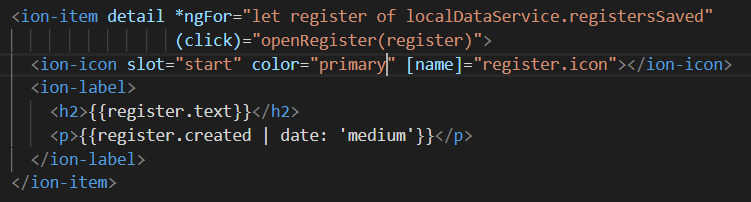
Creamos el header del tab 2  



Ahora importamos el servicio y lo llamamos directamente desde el html



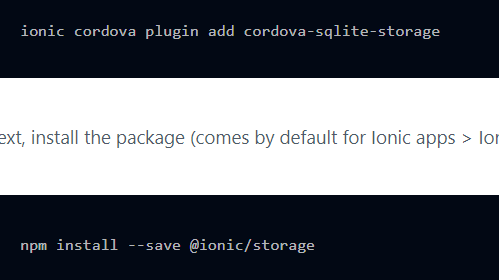


Guardando la información en el storage:

Cargar los registros desde el constructor del servicio

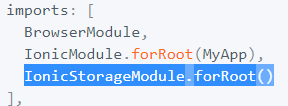
Es necesario instalar el plugin de Storage

<https://ionicframework.com/docs/building/storage>



Y luego agegarlo al app.module en los imports





En el servicio se importa  


Se recorre el arreglo actual para verificar que no exista. De no existir se crea el registro y se guarda en el almacenamiento con el key “register”.

