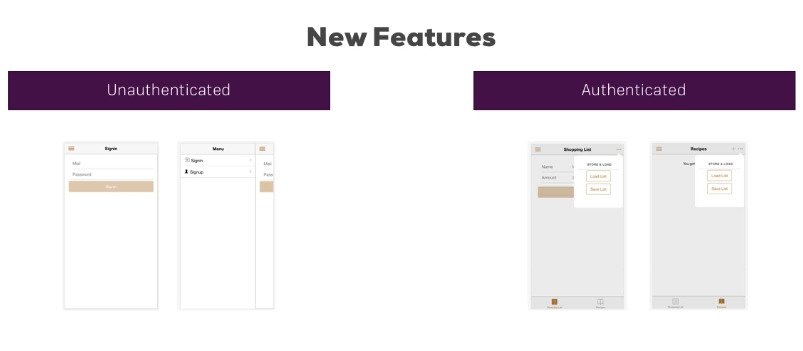
-Añadiendo un sideMenu donde elige si registarrse o iniciar sesión.

-Añadiendole un login a la aplicación de recetas,

-Guardando las recetas y lista de ingredientes en un servidor firebase.



Generar la pagina de signup y signin



Creando el sidemenu desde el app.html

En el app.html:

Se desea que al abrir alguna de las opciones que muestre el sidebar, se abra y desaparezca

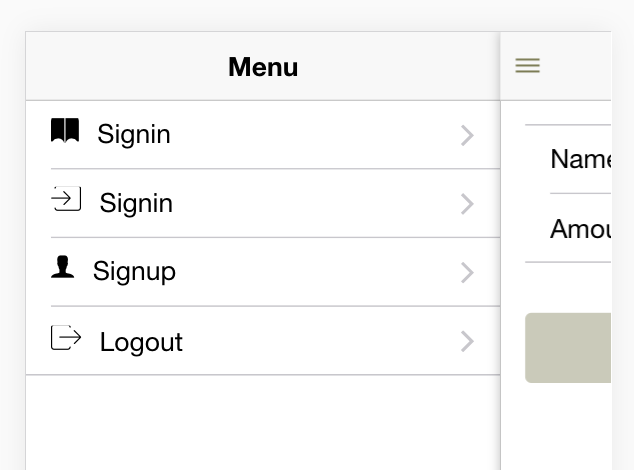
el menú de tabs, por lo que debe sobreescribir el rootPage.

Para esto directamente desde el app.html se agrega el ion-menu.

Añadiendo la propiedad con #nav , lo que hace una referencia local en el ion-nav, y unirlo,

o referir el menú a este tag con [ ].





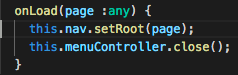
Para poder acceder a la propiedad root:

Se el utiliza el ViewChild y en sus parámetros se hace referencia a la que creamos

anteriormente en el dom:

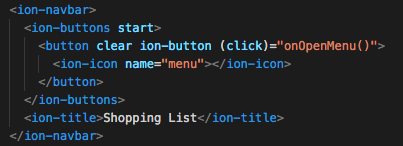


Para luego en la función redirigirlo según la pagina qu se seleccione, previamente declarado como una variable. Es necesario tener acceso al menuController para cerrar el sidemenu ya que al seleccionar una opción, este no cierra automáticamente.



Añadiendo un icono para saber que existe este side menú en las paginas:

En el navbar se agrega un ion-buttons, y dentro un botón que contiene la acción y el icono:



Y luego su respectiva función, que necesita implementar el menú controller desde el constructor



Creando la pagina de Signup y el formulario.

Se crea un formulario con los controles del template, haciendo énfasis en la contraseña utiliza la directiva de minlength =6, ya que firebase acepta solamente contraseñas con esa cantidad, esto puede variar según el backend



Añadiendole el controlador de la plantilla para acceder a los valores y para validar si es valido, luego llamando a una función si detecta el submit que pasa como parámetro el mismo formulario para captar los valores.

El botón de signin se activara solo si el formulario es valido:

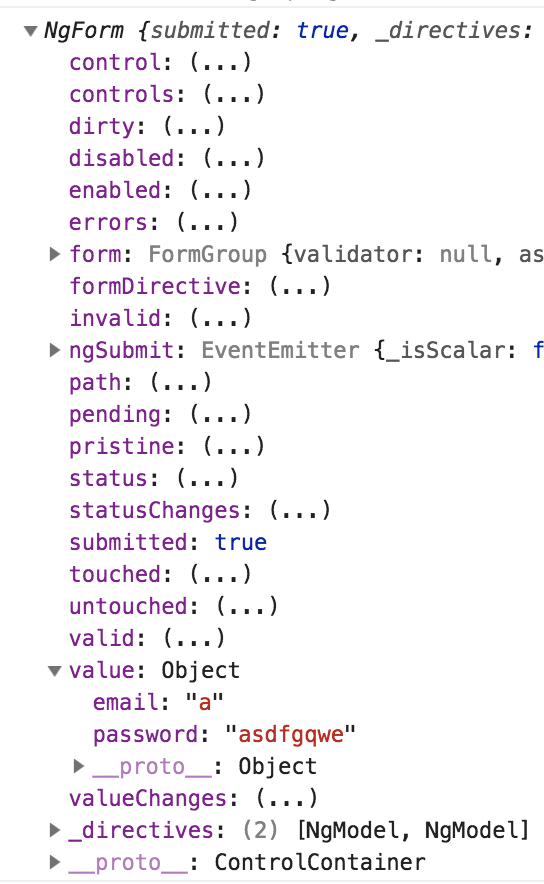




En la función recibimos los datos del formulario, en este caso se declara del tipo NgForm

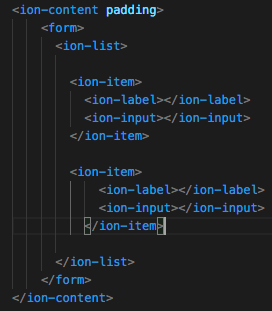


Lo que nos devuelve un objeto, y el campo name resulta relevante ya que asi tendrá el nombre de los datos ingresados en el objeto dentro de value:



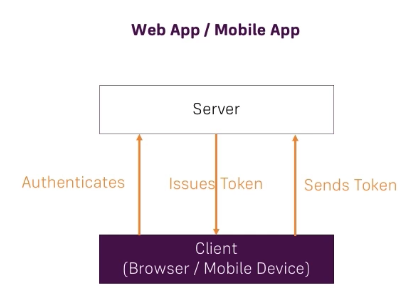
Creando la pagina signin

Siendo esta la estructura:

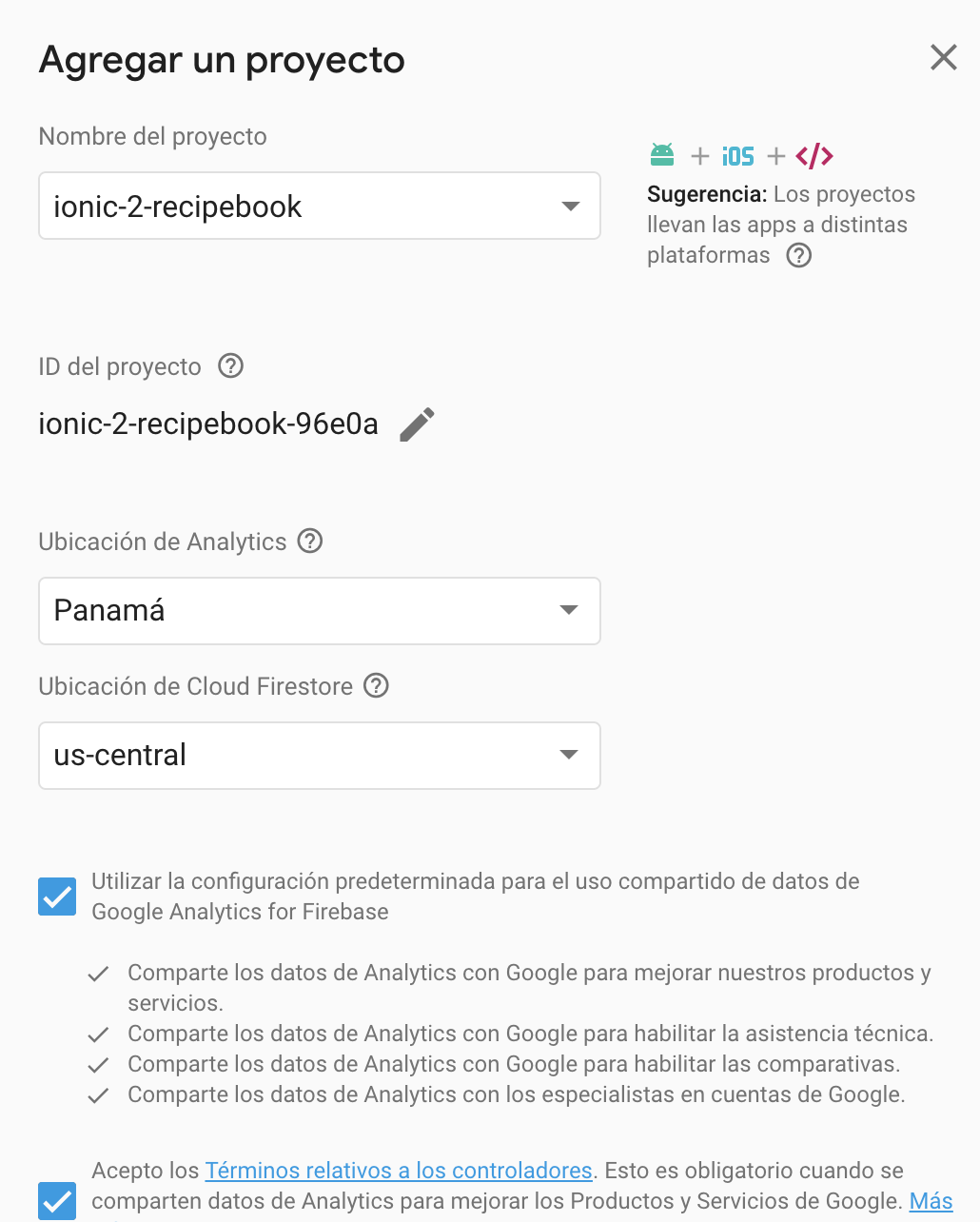


Como trabaja el servicio de autenticación:

Se en via la data que se escribe en el formulario hacia el servidor, y en este se validan los datos. En donde para validar durante la sesión iniciada se envía y recibe un token el cual será evaluado para certificar la validez de la sesión.

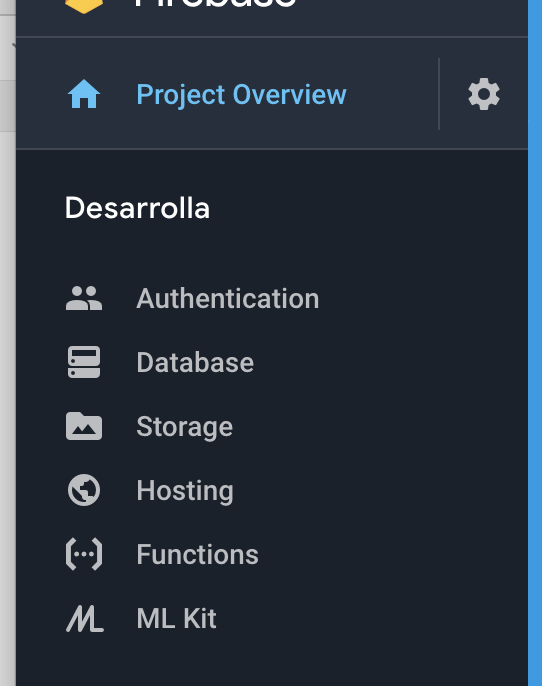


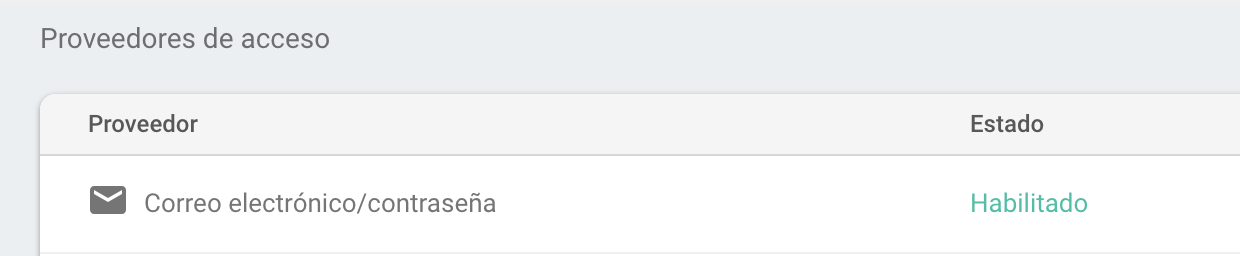
Setting up firebase:



en el panel izquierdo aparece la opción de autenticación:

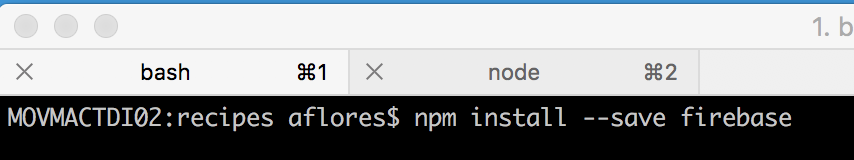
Es necesario tener en cuenta sobre instalar el sdk de firebase.



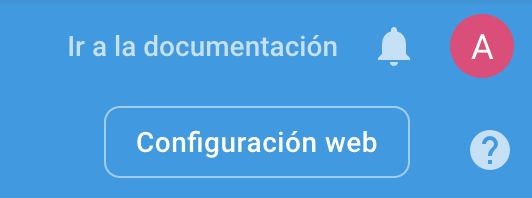


Añadiendo firebase al proyecto:

En la línea de comandos:



Una vez instalado hay que configurarlo, en la pagina en la parte derecha superior:



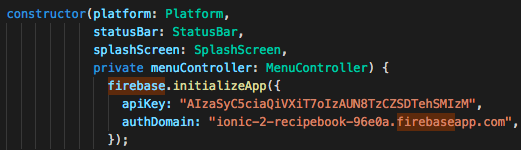
de todo el script que nos arroja copiar únicamente:



Ahora en el app.component:

Importar firebase y dentro del constructor inicializarlo con la información que nos mostrò la configuración de firebase.





IMPORTANTE, LUEGO DE QUE LA PLATAFORMA ESTE LISTA.

Ahora se crea un nuevo servicio auth.service.ts

Se importa firebase:



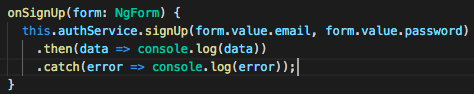
Y dentro los métodos necesarios:

El signUp para crear un nuevo usuario, el cual recibe usuario y contraseña, luego se accede a la función auth y luego createUserAndRetrieveDataWithEmailAndPassword.



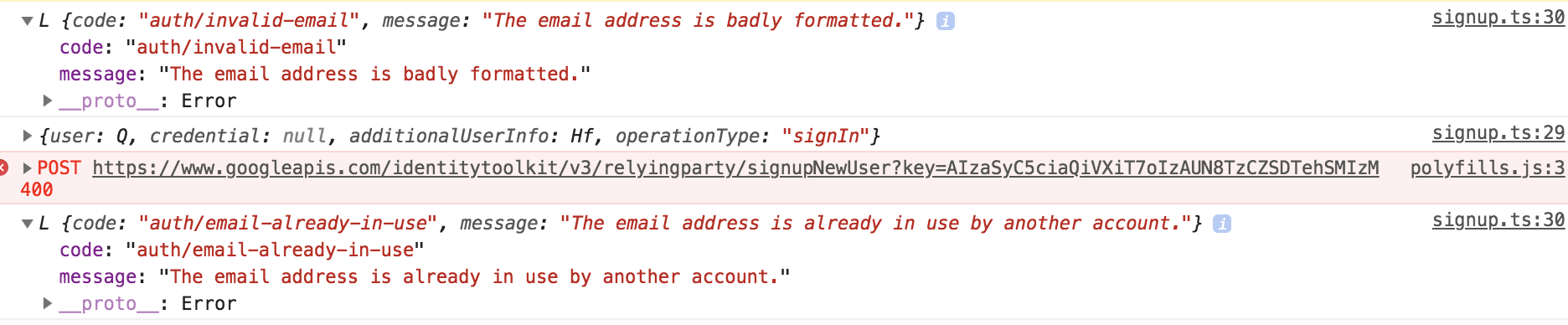
Ahora desde la pagina de registro (signup) se llama a esta función , ya que es una promesa se maneja de la siguiente forma,

El then , va a manejar cuando responda correctamente:

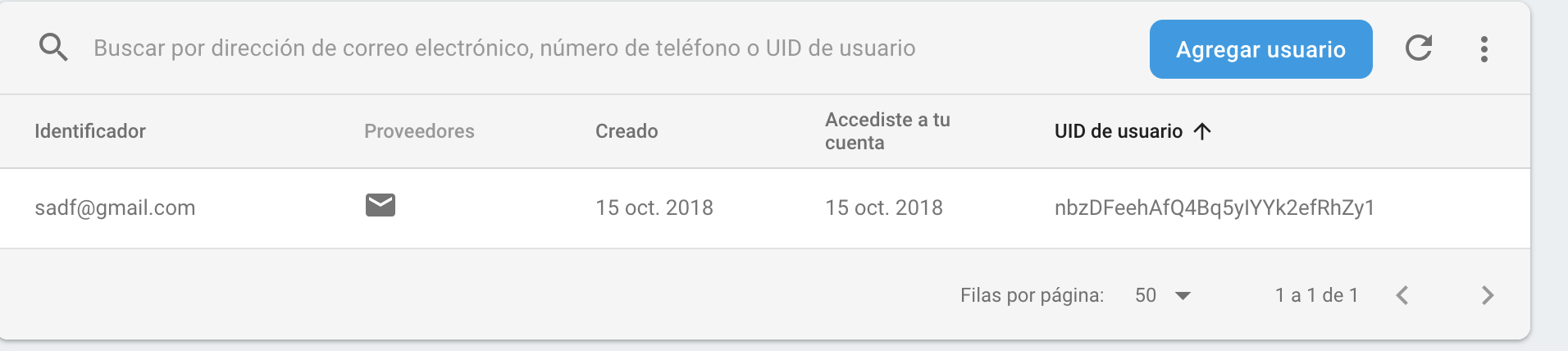


Al introducir la información en el formulario nos da ya validaciones,

1. En caso que el email sea invalido,
2. Correcto
3. Usuario ya registrado



y dentro de firebase



Añadiendo un spinner y mensaje de error:

Ionic cuenta con una librería especial que levanta un loading state.

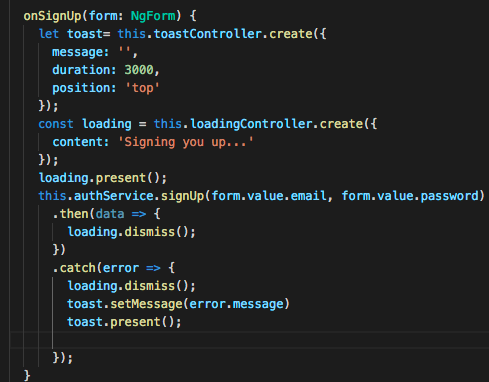
Se importa en el constructor y en el archivo:



Luego dentro de la función que estamos consultando el servicio, se crea una constante loading, que contiene el loading controller con la estructura del mensaje a mostrar.

Se accede a sus propiedades para mostrarlo (present() ) y para cerrarlo una vez retorne el valor ya sea correcto o un error.

Al igual que presentar un toast:

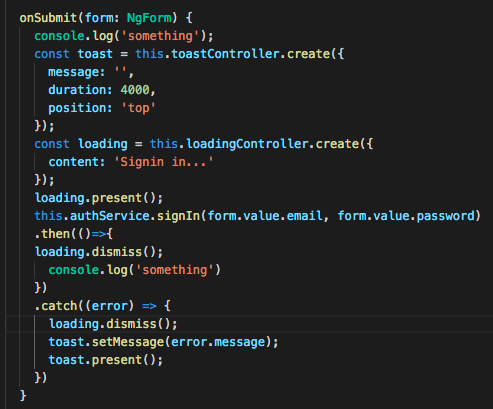


Signin in the app

El en AuthService implementar la función, la que nos retorna una promesa



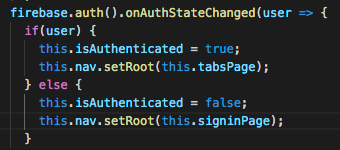
y en el signin component



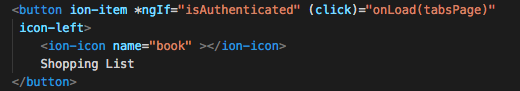
Managing UserState:

Para mostrar una u otra pantalla hay que validar desde el app.component, dentro del constructor:

Llamar a la función onAuthStateChanged, que avisará cuando el estado cambie si esta autenticado o no , recibiendo un objeto promesa.



Ahora una vez ya tenemos la información si el usuario esta autenticado, con la variable del componente isAuthenticated, podemos modificar el hmtl para que muestre las opciones según los provilegios deseados en el momento.



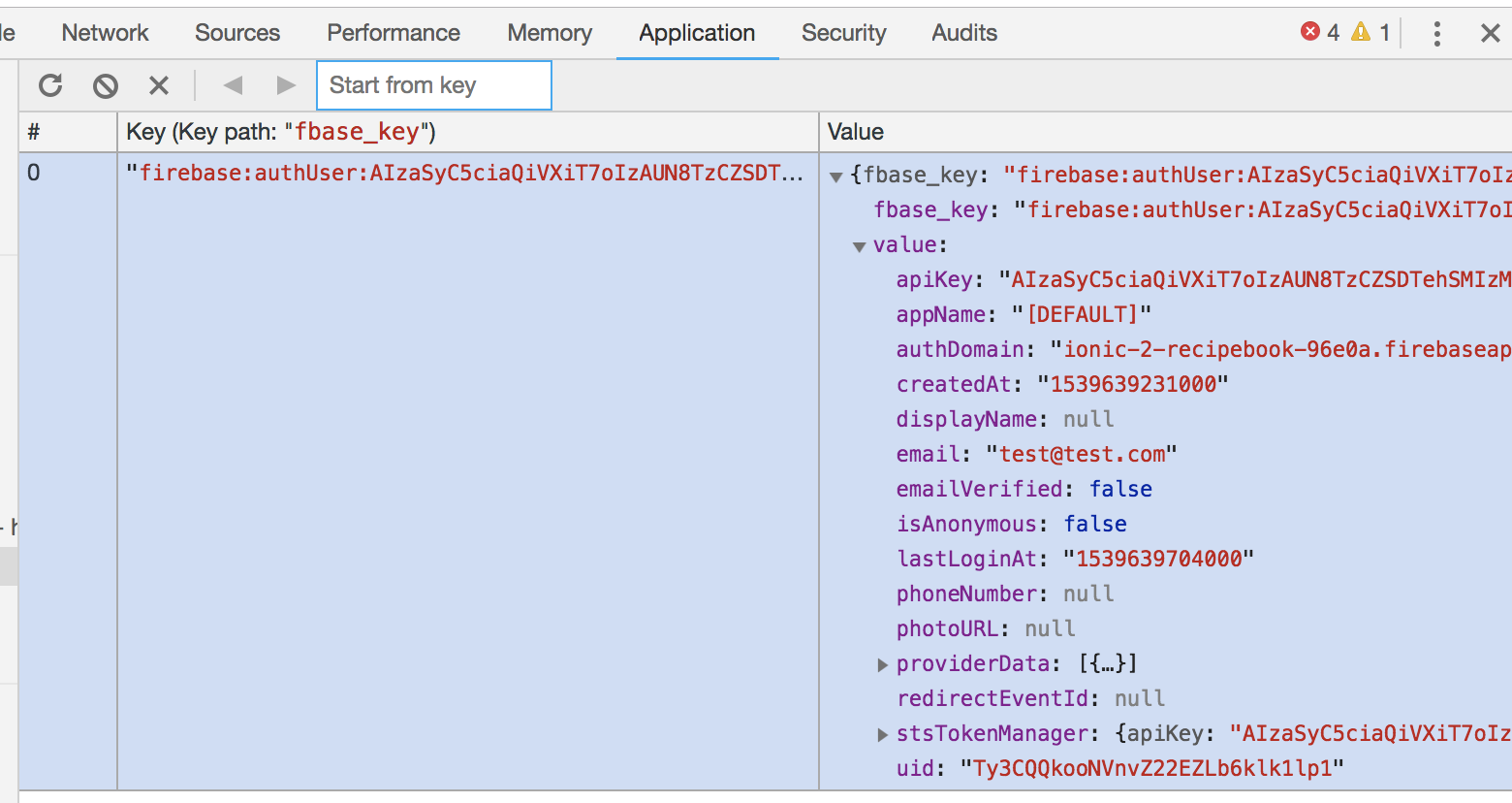
Añadiendo la función de log out

Desde el servicio



Como Firebase almacena el token:

Al iniciar sesión se almacena dentro del local storage, exactamente en :



Añadiendo un popOver Component.

Se crea dentro de la carpeta shopping-list

Sl-options

