Tutorial: BD NoSQL / RealTime con Firebase

Cátedra: Bases de Datos Avanzadas

Dra. Nora Reyes - Mgter Germán Pautsch

2022

En este tutorial veremos el funcionamiento de la sentencia “VALUE” y los pasos que nos permitirán sincronizar un objeto (JSON) en la base de datos.

1. Desde un editor (sugerimos Visual Studio Code) generamos un documento index.html (disponible en el aula) con su estructura básica (HTML5). Luego creamos un bloque con un identificador que luego podamos referenciar.

```
<pre id="objeto"> ↙pre>
```

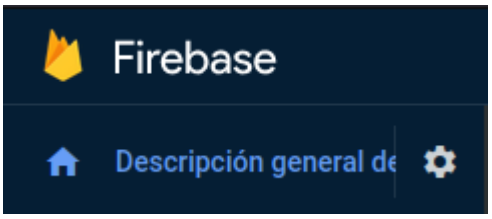
2. Creamos un app.js y lo incluimos en el body del index.html creado en el punto anterior.

```
<script src="app.js">↙script>
```

Una forma de probar si esta vinculación es correcta es poner en el app.js un mensaje para luego visualizarlo en la consola del navegador (Modo inspección):

```
console.log('Entro al app');
```

3. Desde la consola de firebase, Descripción del Proyecto, Configuración del Proyecto, copiamos los datos de conexión para nuestra app (Engranaje en el margen superior derecho).



El código que necesitamos es el de la opción CDN y luce de la siguiente manera:

```
// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyBQstM2yqHw5m2xLxxH_uQI90X_YS4IPUQ",
  authDomain: "ejemplomti2022.firebaseapp.com",
  databaseURL: "https://ejemplomti2022-default-rtdb.firebaseio.com
  projectId: "ejemplomti2022",
  storageBucket: "ejemplomti2022.appspot.com",
  messagingSenderId: "470921170166",
  appId: "1:470921170166:web:7858035de7d18b6b2770d7"
};
```

Solo copiamos y reemplazamos en el código, esta vez en el app.js, la parte sombreada en la imagen anterior.

4. Ahora desde app.js vamos a ver cómo podemos acceder a este elemento desde el DOM (document object model).

```
document.getElementById('objeto')
```

   Acá pasamos la id 'objeto' que creamos en nuestro archivo index.html.

5. Creamos una referencia a la base de datos para que podamos sincronizar el objeto. Vamos a crear esas referencias con dbRefObject = firebase.database().ref(). Esta función ref() es la que nos va a dirigir a la raíz de la base de datos y luego llamamos a child() para acceder a una clave, por ejemplo 'curso'.

6. En firebase/Realtime creamos un objeto con la siguiente estructura:

```
curso: {
  "nombre": "BDA",
  "institucion": "UNNE – UnaM",
  "horas": 15
}
```

7. Para sincronizar los datos en tiempo real utilizamos el método 'on' de la siguiente forma: dbRefObject.on('value', snap). Los parámetros que tenemos que pasarle a este método son: Primero el tipo de evento ('value') para controlar la sincronización desde la base de datos. El segundo parámetro es la función 'snap' que es la que controla el evento. Snap devuelve una foto (snapshot) de la base de datos en ese momento. Esta función será llamada cada vez que haya un cambio en la información del objeto.
   Usando las funciones arrow (=>), en esta primera versión lo que vamos a hacer es devolver el valor del objeto curso en la consola.

8. Podemos cambiar los valores del objeto curso en firebase y observar lo que ocurre. Recuerda que estamos utilizando un evento 'value', cuando llevamos a cabo el cambio no cambia el valor del atributo, sino que lo que cambia es el propio objeto.

   Vamos a modificar el ejemplo para que en lugar de la consola la información se muestre en el bloque <pre> del index.html que creamos inicialmente. Para ello se debe cambiar el código que está después de la flecha de snap por este otro:

```
'snap => {preObject.innerText = JSON.stringify(snap.val(),null, 3);'
```

   Si volvemos al navegador observa cómo la información se muestra ya en la página y cualquier cambio que incluyamos aparece como estás viendo en la app.