

NodeMCU: comunicación con Firebase. Escribe los datos de Firebase en Google Spreadsheet (VII-C).

 agrportfolioeducativo.blogspot.com/2019/07/nodemcu-comunicacion-con-firebase_14.html

Una vez que hemos encontrado la forma de comunicar los dos dispositivos NodeMCU a través de la base de datos Realtime de Firebase cabe preguntarse si es posible mantener un registro de los datos. ¿Qué tal si aprovechamos la hoja de cálculo de la entrada "NodeMCU: proyecto "botón del pánico" (VI-A): la hoja de cálculo y su script." para registrar los datos?

Lo primero que se me puede ocurrir es comunicar Firebase con Google Spreadsheet. Un poco de lectura y averiguo que se usa la clase **UrlFetchApp** para conseguirlo.

El script en google spreadsheet

```
function obtenerDatos(){
// Dirección de la base de datos RealTime de Firebase
var bd="[midireccionenfirebase].firebaseio.com";
// Secreto de la base de datos Firebase
var secreto='ffsfhjrf9sadfnsdfynas7fya78ynf78n7sc';
// Ruta de la base de datos RealTime
var ruta="Estaciones";

    var options = { method:'get', contentType: 'application/json' }; //opciones de la
llamada a fetch
    var firebaseURL = 'https://' + bd + '/' + ruta + '.json?auth='+
secreto; // construyo la ruta

var resultado = UrlFetchApp.fetch(firebaseURL, options);

// Hoja de cálculo activa
var sheet = SpreadsheetApp.getActiveSheet();
var ultimafila = sheet.getLastRow(); // obtiene la última línea

// Fechas y horas actuales.
var fecha=Utilities.formatDate(new Date,"GMT+1","dd/MM/yyyy");
var d =new Date();
```

```
var hora = d.toLocaleTimeString();
```

```
// Objeto JSON según el resultado
```

```
var e = JSON.parse(resultado);
```

```
    for (clave in e) {
```

```
        var columna = clave.charCodeAt(0)-62; // columna de cada valor
```

```
        var valorActual = sheet.getRange(2,columna).getValue();
```

```
        if (valorActual!=e[clave]) { // Si los valores son distintos
```

```
            // 1) Cambio fecha y hora en la fila dos
```

```
            sheet.getRange(2,1).setValue(fecha);
```

```
            sheet.getRange(2,2).setValue(hora);
```

```
            // 2) Cambio valor actual en la fila dos
```

```
            sheet.getRange(2,columna).setValue(parseInt(e[clave]));
```

```
            // 3) Pongo fecha y hora en la última línea
```

```
            sheet.getRange(ultimalinea+1,1).setValue(fecha);
```

```
            sheet.getRange(ultimalinea+1,2).setValue(hora);
```

```
            // 4) Escribo en la última línea un registro de lo que acontece
```

```
            segunActivo(clave,e[clave],ultimalinea+
```

```
1,columna,sheet);
```

```
}
```

```
// Logging de los datos...
```

```
    Logger.log(clave);
```

```
    Logger.log(e[clave]);
```

```
}
```

```
}
```

```
// funcion que escribe en el registro según si está activo o no
```

```
function segunActivo(estacion,dato,fila,columna,sheet) {
```

```
    if (dato==1) {
```

```
        sheet.getRange(fila,columna).setValue("Estación "estacion" se activa");
```

```
        sheet.getRange(fila,columna).setFontColor("darkgreen");
```

```
        sheet.getRange(fila,columna).setBackground("#AAFFAA");
```

```
    } else if (dato==0) {
```

```

sheet.getRange(fila,columna).setValue("Estación "estacion" se desactiva");
sheet.getRange(fila,columna).setFontColor("darkred");
sheet.getRange(fila,columna).setBackground("#FFAAAA");
}
}

```

1. Defino las variables de mi programa: direccion a Firebase, su token o secreto y la ruta a los datos. Si no ponemos la ruta, devuelve un JSON más complicado.
2. La variable **options** recoge las opciones de la llamada que vamos a hacer después con la clase **UrlFetchApp** y el método **fetch**. La variable **firebaseURL** recoge la ruta a los datos. El resultado de aplicar este método se guarda como un string en **resultado**.
3. Obtengo el objeto hoja de cálculo, su última línea, y la fecha y hora actuales.
4. Consigo el objeto **e** como un objeto JSON de la cadena **resultado**. Método **parse**.
5. Por cada **clave** dentro del objeto JSON **e**,
6. Calculo la columna que corresponde a la estación A, B,C...
7. Obtengo el valor actual de la celda,
8. Y si son distintos quiere decir que se ha desactivado o activado la estación luego...
9. Cambio fecha y hora en la fila dos, y también escribo el valor en la celda correspondiente a cada estación en la fila dos.
10. En la siguiente línea vacía del registro escribo la fecha y la hora, y escribo, llamando a la función **segunActivo** el registro de la estación.

Concluyendo: este script, si lo hago funcionar periódicamente (por ejemplo cada minuto), con esa cadencia comprobará si las estaciones están o no activas. Esta funcionalidad no me parece completa, ya que puede pasar que en ese lapso de tiempo la estación se active y se desactive, no registrando nada, o registrando con cierto retraso los sucesos.

La secuencia más correcta sería: cuando se active/desactive una estación, registrar el dato en Firebase y desde el NodeMCU también registrar el dato en la hoja de cálculo. Esto último podríamos hacerlo directamente, como dos sucesos independientes, o bien hacer que se ejecute un script desde el NodeMCU que lea la base de datos Firebase.

=====

Enlaces:

Lee datos desde firebase

<https://stackoverflow.com/questions/25529863/reading-data-from-firebase-with-google-apps-script>

Escribe datos en Firebase

<https://www.sohamkamani.com/blog/2017/03/09/sync-data-between-google-sheets-and-firebase/>