



Firestore Functions

with Typescript

🏠 **Desarrollo** enfocado en **Angular** en remoto.

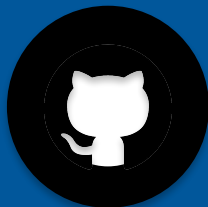
👤🏢 **Instructor** en **Udemy** con +35K estudiantes.

📖 **Reviewer** en **packtpub**:

- 📖 Learning Typescript 2.x: second edition.

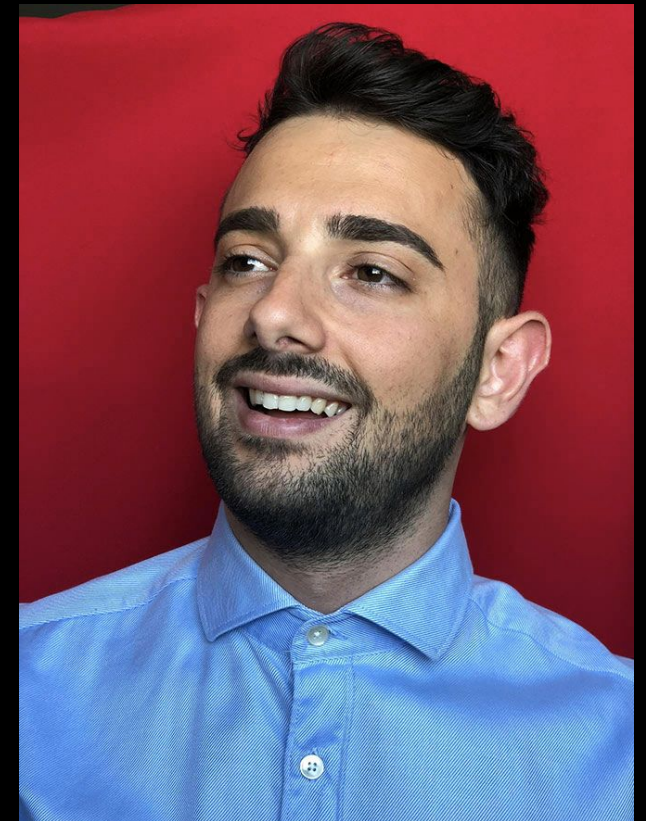
- 📖 Typescript 3.0 Quick Start Guide.

☐ Contacta conmigo en los siguientes links.



Alejandro Lora

Freelance Software
Developer



Agenda

Introducción a Firebase

- ¿Qué es Firebase?
- Planes de precios
- Firebase Functions
- Firebase Cloud Firestore
- Instalación SDK

Creando Proyecto

- Crear proyecto Firebase Functions
- Estructura proyecto
- Hello world
- Añadiendo proyecto en console
- Debugging en local
- Hot reloading
- Creando documento y colección en Firestore

Deployment

- Deployment
- Actualizando Cloud Function
- Activadores de Eventos Cloud Firestore
- Añadiendo Cron-Job

TS

Introducción a Firebase









¿QUÉ ES FIREBASE?

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

Firestore es una plataforma para el desarrollo de aplicaciones webs y móviles desarrollada por James Tamplin y Andrew Lee en 2012 y adquirida por Google en 2014.

	Android reference View full API now		iOS reference View full API for Swift & Obj-C
	Web reference View full API for JS Client		C++ reference View full API now
	Unity reference View full API now		Admin reference View full API now

PLAN DE PRECIOS

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

Plan Spark

Límites generosos para aficionados

Sin cargo

Plan Flame

Precios predecibles para apps en expansión

USD 25/mes

Plan Blaze

[Calcula los precios de las apps a gran escala](#)

Pago por uso

✓ Free usage from Spark plan included*

PLAN SPARK (FREE)

- Real time Database
- Cloud Firestore
- Storage
- Cloud Functions

- Phone Auth
- Hosting
- Test Lab
- ML Kit

Document writes = 20K/día

Stored data = 1GB

Document reads = 50K/día

Bandwidth = 10GB/mes

Document deletes =
20K/día

FIREBASE FUNCTIONS

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

Firestore Cloud Functions te permite ejecutar de forma automática el código de backend en respuesta a eventos activados por las funciones de Firestore y/o solicitudes HTTPS.

Si la función está ocupada con muchos eventos, Google crea más instancias para controlar el trabajo con más rapidez. Si la función está inactiva, las instancias se borran.

Cuando el desarrollador actualiza la función mediante la implementación del código actualizado, todas las instancias de la versión antigua se borran y se reemplazan por instancias nuevas.

Cuando un desarrollador borra la función, se borran todas las instancias y se quita la conexión entre la función y el proveedor de eventos.

FIREBASE CLOUD FIRESTORE

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

Cloud Firestore es una base de datos NoSQL flexible, escalable y en la nube, a fin de almacenar y sincronizar datos para la programación en el lado del cliente y del servidor.

Es flexible y escalable para la programación en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud Platform.

Al igual que Firebase Realtime Database, mantiene tus datos sincronizados entre apps cliente a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión para dispositivos móviles y la Web.

Las consultas de Cloud Firestore son expresivas, eficientes y flexibles, aunque la herramienta sigue en fase Beta.

INSTALACIÓN SDK

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

```
npm install -g firebase-tools
```

```
alex@MBPdeAlejandro ➤ ~/Desktop/firebase-functions-typescript ➤ npm install -g firebase-tools
```

TS

Creando Proyecto



AÑADIENDO PROYECTO A CONSOLE

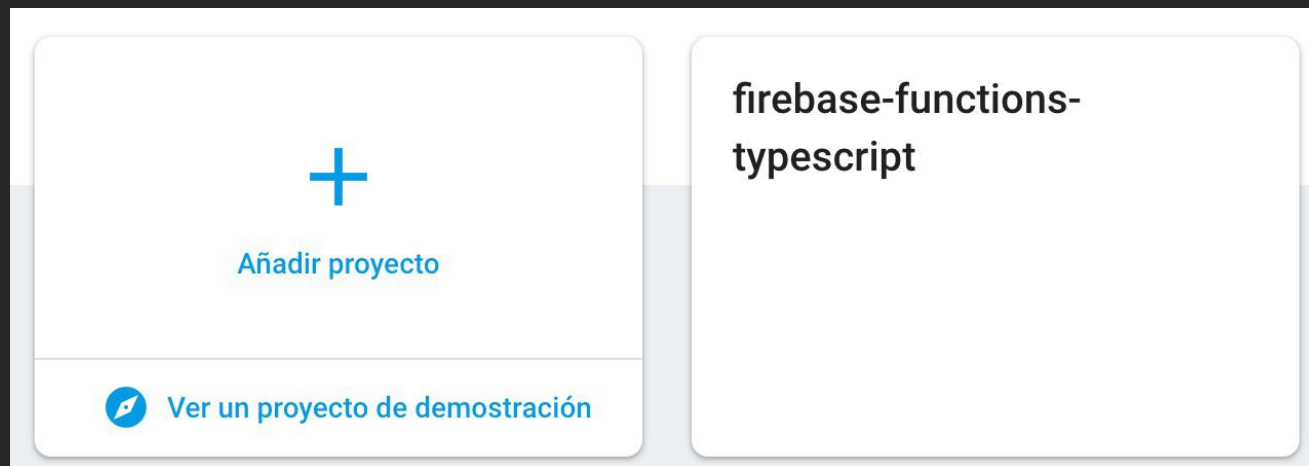
INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

Crear cuenta en Firebase y acceder a la consola.

<https://console.firebase.google.com/>



CREAR PROYECTO FIREBASE FUNCTIONS

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

```
alex@MBPdeAlejandro ~/Desktop/firebase-functions-typescript$ firebase login
Already logged in as alexandrofpo@gmail.com
alex@MBPdeAlejandro ~/Desktop/firebase-functions-typescript$ firebase init functions
```

```
#####  #####  #####  #####  #####  #####  #####
##      ##      ##      ##      ##      ##      ##
#####  ##  #####  #####  #####  #####  #####
##      ##      ##      ##      ##      ##      ##
##      ##      ##  #####  #####  ##      ##  #####
```

You're about to initialize a Firebase project in this directory:

`/Users/alex/Desktop/firebase-functions-typescript`

=== Project Setup

First, let's associate this project directory with a Firebase project. You can create multiple project aliases by running `firebase use --add`, but for now we'll just set up a default project.

```
? Select a default Firebase project for this directory: fir-functions-typescript-94df0 (firebase-functions-typescript)
1 Using project fir-functions-typescript-94df0 (firebase-functions-typescript)
```

=== Functions Setup

A `functions` directory will be created in your project with a Node.js package pre-configured. Functions can be deployed with `firebase deploy`.

```
? What language would you like to use to write Cloud Functions? TypeScript
? Do you want to use ESLint to catch probable bugs and enforce style? Yes
✓ Wrote functions/package.json
✓ Wrote functions/tslint.json
✓ Wrote functions/tsconfig.json
✓ Wrote functions/src/index.ts
✓ Wrote functions/.gitignore
? Do you want to install dependencies with npm now? Yes
```

```
$ firebase login
```

```
$ firebase init functions
```













ESTRUCTURA PROYECTO

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

▲ FIREBASE-FUNCTIONS-TYPESCRIPT

- ▲  functions
 - ▶  node_modules
 - ▲  src
 -  index.ts
 -  .gitignore
 -  package-lock.json
 -  package.json
 -  tsconfig.json
 -  tslint.json
 -  .firebaseerc
 -  .gitignore
 -  firebase.json

HELLO WORLD

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

TS index.ts x

functions ▸ src ▸ TS index.ts ▸ ...

```
1  import * as functions from "firebase-functions"; 57K (gzipped: 17.7K)
2
3  // Start writing Firebase Functions
4  // https://firebase.google.com/docs/functions/typescript
5
6  export const helloWorld = functions.https.onRequest((request, response) => {
7    |   response.send("Hello from Firebase!");
8  | });
9
```

DEBUGGING EN LOCAL

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

- 1) `$ npm install -g @google-cloud/functions-emulator` // Emulador para debugging
- 2) `$ firebase list` ⇒ `firebase use -add [ID]` // Lista proyectos y selecciona
- 3) `$ functions-emulator start` // Arranca emulador
- 4) `$ functions-emulator inspect helloWorld` // Emulator attached (see port = 9229)
- 5) Crear configuración para debugging y lanzar // VSCode launch.json (use same port)
- 6) Añadir breakpoint



DEBUGGING EN LOCAL

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

- 1) `$ npm install -g @google-cloud/functions-emulator` // Emulador para debugging
- 2) `$ firebase list` ⇒ `firebase use -add [ID]` // Lista proyectos y selecciona
- 3) `$ functions-emulator start` // Arranca emulador
- 4) `$ functions-emulator inspect helloWorld` // Emulator attached (see port = 9229)
- 5) Crear configuración para debugging y lanzar // VSCode launch.json (use same port)
- 6) Añadir breakpoint
- 7) `$ functions-emulator call helloWorld`

```
{  
  "type": "node",  
  "request": "attach",  
  "name": "Attach to Function",  
  "port": 9229  
}
```

```
"run:emulator": "functions-emulator start",  
"inspect:helloWorld": "functions-emulator inspect helloWorld",  
"call:helloWorld": "functions-emulator call helloWorld"
```

HOT RELOADING

INTRODUCCIÓN FIREBASE

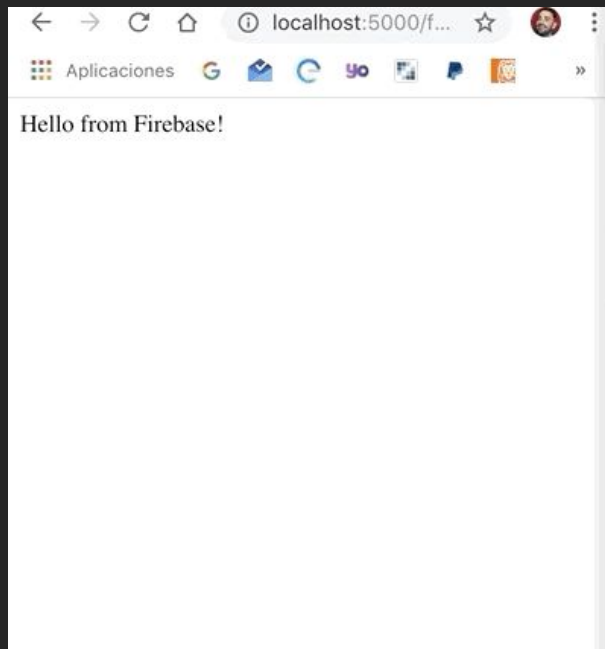
CREANDO PROYECTO

DEPLOYMENT

```
$ npm install concurrently
```

```
$ npm run dev
```

```
"dev": "concurrently \"tsc --watch\" \"firebase serve --only functions\"",  
"run:emulator": "functions-emulator start",  
"inspect:helloWorld": "functions-emulator inspect helloWorld",  
"call:helloWorld": "functions-emulator call helloWorld"
```



```
src > TS index.ts > helloWorld  
1 // The Cloud Functions for Firebase SDK to create Cloud Functions and setup triggers.  
2 import * as functions from "firebase-functions"; 57K (gzipped: 17.7K)  
3  
4 export const helloWorld = functions.https.onRequest(async (request, response) => {  
5   response.send(`Hello from Firebase!`);  
6 });  
7
```

DOCUMENTO Y COLECCIÓN FIRESTORE

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

The screenshot shows the Firebase console interface. On the left is a sidebar with the 'Firebase' logo and navigation links: 'Project Overview', 'Desarrollo' (with sub-links for Authentication, Database, Storage, Hosting, Functions, and ML Kit), 'Calidad' (with links for Crashlytics, Performance, and Test Lab), 'Analíticas' (with links for Dashboard, Events, Conversions, and Attribution), and 'Crecimiento' (with links for Predictions, A/B Testing, and Cloud Messaging). The main area displays the 'firebase-functions-typescript' project with a 'Beta' badge and the 'Cloud Firestore' title. A modal dialog titled 'Reglas de seguridad de Cloud Firestore' is open, explaining that security rules must be defined. It offers two options: 'Empezar con el modo de bloqueo' (selected) and 'Empezar con el modo de prueba'. A code block shows a sample security rule that denies all reads and writes. A warning message states that all third-party operations will be rejected. At the bottom, a note mentions that Cloud Datastore cannot be used in this project, and there are 'Cancelar' and 'Habilitar' buttons.

firebase-functions-typescript

Beta

Cloud Firestore

Reglas de seguridad de Cloud Firestore

Cuando defines tu estructura de datos, tendrás que escribir reglas para protegerlos.
[Más información](#)

- ☒ **Empezar con el modo de bloqueo**
Rechaza todas las operaciones de lectura y escritura para que tu base de datos sea privada
- ☐ **Empezar con el modo de prueba**
Permite todas las operaciones de lectura y escritura en tu base de datos para configurarla rápidamente

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

i Se rechazarán todas las operaciones de lectura y escritura de terceros

Si habilitas Cloud Firestore (Beta), no podrás usar Cloud Datastore en este proyecto, en particular desde la aplicación de App Engine asociada.

Cancelar **Habilitar**

DOCUMENTO Y COLLECCIÓN FIRESTORE

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

TS index.ts x

```
src ▶ TS index.ts ▶ ...
1 // The Cloud Functions for Firebase SDK to create Cloud Functions and setup triggers.
2 import * as functions from "firebase-functions"; 57K (gzipped: 17.7K)
3 // The Firebase Admin SDK to access the Firebase Realtime Database.
4 import * as admin from "firebase-admin";
5
6 const app = admin.initializeApp();
7
8 export const helloWorld = functions.https.onRequest(async (request, response) => {
9   const now = new Date();
10  const ref = admin.firestore(app).collection("test");
11  ref.add({ date: now }).then(() => {
12    const date = getDateAsString(now);
13    response.send(`Date added! ${date.day}/${date.month}/${date.year}:${now.getMilliseconds()}`);
14  });
15 });
16
17 function getDateAsString(date: Date): { day: string; month: string; year: string } {
18   const day = getDayString(date);
19   const month = getMonthString(date);
20   const year = getYearString(date);
21   return { day, month, year };
22 }
23
24 function getDayString(date: Date): string {
25   const dd = date.getDate();
26   return dd < 9 ? "0" + dd : dd.toString();
27 }
28
29 function getMonthString(date: Date): string {
30   const mm = date.getMonth() + 1;
31   return mm < 9 ? "0" + mm : mm.toString();
32 }
33
34 function getYearString(date: Date): string {
35   return date.getFullYear().toString();
36 }
```

1) Añadimos admin SDK

2) Inicializamos APP

3) Referencia a collección

4) Añadimos documento

5) Añadimos funciones de ayuda

TS

Deployment



DEPLOYMENT

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

```
$ npm run deploy // Ejecuta predeploy tasks definidas en firebase.json (lint & tsc)
```

```
ERROR: ../functions/src/index.ts[11, 3]: Promises must be handled appropriately
```

Promise style

```
export const helloWorld = functions.https.onRequest((request, response) => {
  const now = new Date();
  const ref = admin.firestore(app).collection("test");
  return ref.add({ date: now }).then(() => {
    const date = getDateAsString(now);
    response.send(`Date added! ${date.day}/${date.month}/${date.year}`);
  });
});
```

Async/Await style

```
export const helloWorld = functions.https.onRequest(async (request, response) => {
  const now = new Date();
  const ref = admin.firestore(app).collection("test");
  await ref.add({ date: now });
  const date = getDateAsString(now);
  response.send(`Date added! ${date.day}/${date.month}/${date.year}`);
});
```

✓ Deploy complete!

Project Console: <https://console.firebase.google.com/project/fir-functions-typescript-94df0/overview>



DEPLOYMENT

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT






firebase-functions-typescript ▾

























Ir a la documentación  

Functions

Panel de control Estado Registros Uso

Función	Activador	Región	Tiempo de ejecución	Memoria	Tiempo de espera
helloWorld	HTTP Solicitud https://us-central1-fir-functions-typescript-94df0.cloudfunctions.net/helloWorld	us-central1	Node.js 6	256 MB	60s

     <https://us-central1-fir-functions-typescript-94df0.cloudfunctions.net/helloWorld>

Aplicaciones                        

Date added! 08/12/2018:684

DEPLOYMENT

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

firebase-functions-typescript

Ir a la documentación

Database

Cloud Firestore BETA

?

Datos

Reglas

Índices

Uso

> test > kYcH9MfSjMD4...

fir-functions-typescript-94df0	test	kYcH9MfSjMD4qsK6GrAJ
+ Añadir colección	+ Añadir documento	+ Añadir colección
test >	kYcH9MfSjMD4qsK6GrAJ >	+ Añadir campo
		date: 8 de diciembre de 2018, 13:14:07 UTC+1

ACTUALIZANDO CLOUD FUNCTION

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

TS index.ts x

src ▶ TS index.ts ▶ ...

```
1 // The Cloud Functions for Firebase SDK to create Cloud Functions and setup triggers.
2 import * as functions from "firebase-functions"; 57K (gzipped: 17.7K)
3 // The Firebase Admin SDK to access the Firebase Realtime Database.
4 import * as admin from "firebase-admin";
5
6 import { getDateAsString } from "../helper";
7
8 const app = admin.initializeApp();
9
10 export const helloWorld = functions.https.onRequest(async (request, response) => {
11   response.send(`¡Hola GDG Sevilla!`);
12 });
13
14 export const ping = functions.https.onRequest((request, response) => {
15   response.send(200);
16 });
17
18 export const dateTest = functions.https.onRequest(async (request, response) => {
19   const now = new Date();
20   const ref = admin.firestore(app).collection("test");
21   await ref.add({ date: now });
22   const date = getDateAsString(now);
23   response.send(`Date added! ${date.day}/${date.month}/${date.year}:${now.getMilliseconds()}`);
24 });
```

- 1) Funciones en helper.ts
- 2) Refactor **helloWorld** function
- 3) Crear **ping** function
- 4) Renombrar **dateTest** function
- 5) Desplegamos de nuevo

ACTUALIZANDO CLOUD FUNCTION

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

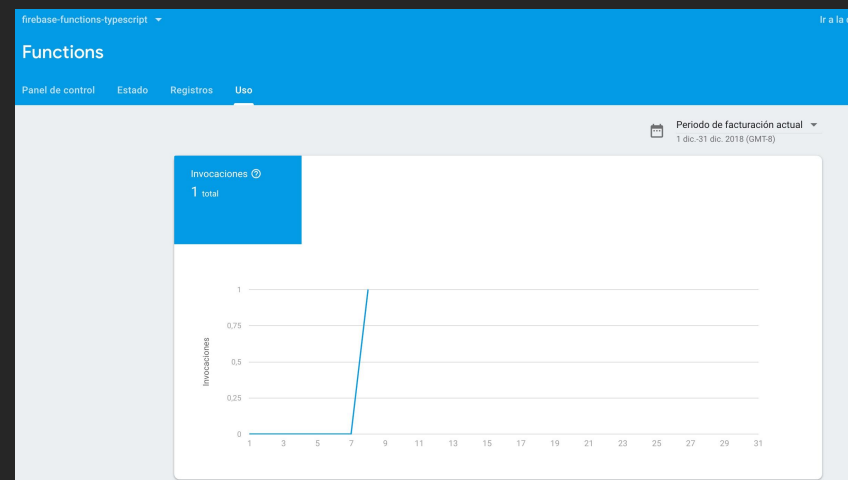
DEPLOYMENT

firebase-functions-typescript ▾ Ir a la documentación 🔔 👤

Functions ?

Panel de control Estado Registros Uso

Función	Activador	Región	Tiempo de ejecución	Memoria	Tiempo de espera
dateTest	HTTP Solicitud https://us-central1-fir-functions-typescript-94df0.cloudfunctions.net/dateTest	us-central1	Node.js 6	256 MB	60s
helloWorld	HTTP Solicitud https://us-central1-fir-functions-typescript-94df0.cloudfunctions.net/helloWorld	us-central1	Node.js 6	256 MB	60s
ping	HTTP Solicitud https://us-central1-fir-functions-typescript-94df0.cloudfunctions.net/ping	us-central1	Node.js 6	256 MB	60s



ACTIVADORES DE EVENTOS CLOUD FUNCTION

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

```
26 exports.createDateEvt = functions.firestore.document("test/{id}").onCreate((snap, context) => {
27     const timestamp = context.timestamp;
28     const authType = context.authType;
29     const eventId = context.eventId;
30     const eventType = context.eventType;
31     const resource = context.resource;
32
33     const newValue = snap.data();
34     const value = newValue.date;
35
36     console.log(`CREATED DATE EVENT: [timestamp: ${timestamp}] - [authType: ${authType}] - [eventI
37     console.error(`CREATED DATE VALUE: [value: ${value}] - [snap: ${JSON.stringify(newValue)}]`);
38 });
```

ACTIVADORES DE EVENTOS CLOUD FUNCTION

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

firebase-functions-typescript

Ir a la documentación

Functions

Panel de control

Estado

Registros

Uso

Registros de búsqueda

Todas las funciones

Todos los niveles de registro

Hora

Nivel

Función

Mensaje de evento

8 dic. 2018

1:51:55.491 p. m.

createDateEvt

Function execution took 1657 ms, finished with status: 'ok'

1:51:55.399 p. m.

createDateEvt

Function returned undefined, expected Promise or value

1:51:55.399 p. m.

createDateEvt

CREATED DATE VALUE: [value: [object Object]] - [snap: {"date":{"_seconds":1544273503,"_nanoseconds":763000000}}]

1:51:55.389 p. m.

createDateEvt

▶ CREATED DATE EVENT: [timestamp: 2018-12-08T12:51:53.250575Z] - [authType: undefined] - [eventId: 17a12c52-62a9-48b4-9ec0-11b395...

1:51:53.835 p. m.

createDateEvt

▶ Billing account not configured. External network is not accessible and quotas are severely limited. Configure billing account t...

1:51:53.834 p. m.

createDateEvt

Function execution started

ACTIVADORES DE EVENTOS CLOUD FUNCTION

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

firebase-functions-typescript ▾

[Ir a la documentación](#)



Functions







Panel de control

Estado

Registros

Uso

Función	Activador	Región	Tiempo de ejecución	Memoria	Tiempo de espera
createDateEvt	 document.create /test/{id}	us-central1	Node.js 6	256 MB	60s
dateTest	 Solicitud https://us-central1-fir-functions-typescript-94df0.cloudfunctions.net/dateTest	us-central1	Node.js 6	256 MB	60s
helloWorld	 Solicitud https://us-central1-fir-functions-typescript-94df0.cloudfunctions.net/helloWorld	us-central1	Node.js 6	256 MB	60s
ping	 Solicitud https://us-central1-fir-functions-typescript-94df0.cloudfunctions.net/ping	us-central1	Node.js 6	256 MB	60s


AÑADIENDO CRON JOB

INTRODUCCIÓN FIREBASE

CREANDO PROYECTO

DEPLOYMENT

<https://cron-job.org/en/>







Welcome Signup **Members** FAQ

Members **Cronjobs** Statistics Account

Logged in as Alejandro Lora. [Log out](#)

Cronjobs

 Create cronjob

	Title, Address	Last execution	Next execution	
	Firebase Function by Typescript https://us-central1-fir-functions-types...	Today, 15:28:36 Successful (11,5 s)	Today, 15:32	<div> History  Edit</div>

TS

¿PREGUNTAS?

¡ MUCHAS GRACIAS !

