

Cloud Functions to the Rescue!

Zamzam Farzamipooya
Senior iOS Engineer & Tech Lead @ Veo
FrenchKit 2022
Paris

$\sqrt{123}$

"The functions of a good software is to make
the complex appear to be simple! !"

-Grady Booch



+ x ÷

Common Mobile Engineer's Pains



"It's JUST calling 4 APIs for loading one page; why are you making such a fuss about it?!"

Number of API Calls



"Can't you just handle that part of the logic client side? we don't have time to add that now!"

Handling business logic



"There is this user settings that needs to be saved, just do it on the client side!"

Keeping local data



How many of you have wished to be able to do some
backend work to avoid doing these things client side?





Usual Blockers



01. Don't know where to start.

02. Don't know what tech stack to choose



03. Don't know how to deploy or maintain a backend project

04. It all seems way too MUCH work to learn and start





Cloud Functions to the Rescue





Cloud Functions



Serverless framework

Runs on event triggers

Code is stored in GCP

Zero maintenance

Private and secure

Deploy with one line

Integrates with FCM

Can run intensive tasks

Integrates with third-party services and APIs





Cloud Functions V2



Can execute more than one request at a time

New trigger types like Crashlytics velocity alerts

Http functions can have a 1 hour time out



Quick Start - Google Cloud Panel

You can make your first function in **5 minutes** through Google Cloud Panel!

The screenshot shows a browser window with the title "Functions – Cloud Functions". The main content area is titled "Tutorial" and "Step 2 of 5". It displays a numbered list of steps for creating a function:

1. In the Google Cloud console navigation menu, click **Cloud Functions**. You can see where it is by clicking the following button:
2. On the Functions overview page, click **Create function**. [Show me](#)
3. In **Function name**, enter the following name:
4. In **HTTP Trigger type**, select **HTTP**.
5. Select [Allow unauthenticated invocations](#).
6. Click **Save**, and then click **Next**.
7. In **Runtime**, select your preferred language.
8. In **Source code**, select **Inline editor** and keep the default function that is provided in the inline editor, which appears next to the **Source code** list.
9. Click **Deploy**.

Your new function is listed on the Functions overview page. When the deployment is complete, a check mark appears next to the function.

To learn how to test the deployed function, click **Next**.

Navigation buttons at the bottom: PREVIOUS and NEXT.

Firebase CLI

Create Firebase
project

Install node.js &
Firebase CLI

Initialize your project
with JavaScript or
TypeScript



```
$ npm install -g firebase-tools  
$ firebase login  
$ firebase init functions
```

Write Your
Functions



Ship It! Ship It! Ship It!

1

Write your code

Index.ts or index.js
is the heart of your
functions

2

Test it

You can easily run your
functions locally and test it
in the Firebase Local
Emulator Suite

3

Deploy it

You can deploy it with
one line of code!

Project Structure Using TypeScript

Project/

* functions/ # Directory containing functions code

*

* Source/ # Directory containing TypeScript code

* index.ts # Main source file for your code



```
6 export const userIsLegal = functions.https  
7   .onCall((data, context) => {  
8     const name = data.name;  
9     const familyName = data.familyName;  
10    const birthYear = data.birthYear;
```

HTTPS callable functions

Data sent from the clients

```
12 // Error Handling  
13 if (!(typeof name === "string") || name.length === 0) {  
14   // Throwing an HttpsError so that the client gets the error details.  
15   throw new functions.https.HttpsError("invalid-argument",  
16     "The function must be called with \"name\" argument");  
17 }
```

Error handling

```
19 // Business logic  
20 const isLegal = hasLegalAge(birthYear);
```

Add your business logic

```
22 return {  
23   fullName: `${name} ${familyName}`,  
24   isLegal: isLegal,  
25 };  
26 })};
```

Return the result



Let's Test It



01. *Save your code*



Your IDE might not have
autosave like Xcode :D



02. \$ cd .../functions

Navigate to your
functions folder



03. \$ npm run serve



In your package.json file
you can see this line

```
"serve": "npm run build &&  
firebase emulators:start",
```



Console Log

```
i ui: Emulator UI logging to ui-debug.log
i functions: Watching "/Users/zamzampooya/Documents/Apps/FKCloudFunctions/functions" for Cloud Functions...
✓ functions[us-central1-userIsLegal]: http function initialized (http://localhost:5001/halogen-basis-356610/us-central1/userIsLegal)
```

```
✓ All emulators ready! It is now safe to connect your app.
i View Emulator UI at http://localhost:4042
```

Emulator	Host:Port	View in Emulator UI
Functions	localhost:5001	http://localhost:4042/functions
Firestore	0.0.0.0:8082	http://localhost:4042/firestore

Emulator Hub running at localhost:4400

Other reserved ports: 4500

Issues? Report them at <https://github.com/firebase/firebase-tools/issues> and attach the *-debug.log files.

i Remember this is a local environment. Visit the production version of [halogen-basis-356610](#) in the console.

[View project](#) [Dismiss](#)

Emulator overview

Authentication emulator

Status
Off ∅

Port number
N/A

[Go to emulator](#)

Firebase emulator

Status
On ✓

Port number
8082

[Go to emulator](#)

Realtime Database emulator

Status
Off ∅

Port number
N/A

[Go to emulator](#)

Cloud Functions emulator

Status
On ✓

Port number
5001

[View logs](#)

Storage emulator

Status
Off ∅

Port number
N/A

[Storage](#)

Cloud Hosting emulator

Status
Off ∅

Port number
N/A

PubSub emulator

Status
Off ∅

Port number
N/A

Cloud Extensions emulator

Status
Off ∅

Port number
N/A

Emulator UI

POST

http://localhost:5001/halogen-basis-356610/us-central1/userIsLegal

Send

▼

Cookies

Beautify

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings
 none form-data x-www-form-urlencoded raw binary GraphQL **JSON** ▾

```
1 [ {  
2   "data": {  
3     "name": "Zamzam",  
4     "familyName": "Pooya",  
5     "birthYear": 1993  
6   }  
7 }
```

Test It (With Postman)

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 127 ms Size: 302 B

Save Response ▾

Pretty Raw Preview Visualize JSON ▾

```
1 [ {  
2   "result": {  
3     "fullName": "Zamzam Pooya",  
4     "isLegal": true  
5   }  
6 }
```



Call Functions From Mobile Apps



Call functions via
Https requests



Call functions from
your app with Cloud
Functions SDK



Let's Code



```
8 import Foundation
9 import FirebaseFunctions
10
11 class CallableFunctionsManager: ObservableObject {
12
13     lazy var functions = Functions.functions()
14
15     func callFunction(requestModel: IsLegalRequestModel) async throws {
16         let userIsLegal = functions.httpsCallable("userIsLegal",
17                                         requestAs: IsLegalRequestModel.self,
18                                         responseAs: IsLegalResponseModel.self)
19
20         let result = try await userIsLegal(requestModel)
21
22         print("\(result.fullName) has legal age: \(result.isLegal)")
23     }
24
25 }
26
27 struct IsLegalRequestModel: Encodable {
28     var name: String
29     var birthYear: Int
30 }
31
32 struct IsLegalResponseModel: Decodable {
33     var fullName: String
34     var isLegal: Bool
35 }
```

Importing the SDK

Call it like a
normal swift
function

Using Codable models for
Request and Response

```
metadata.emulator.name="functions"

22:38:53 I functions Watching "/Users/zamzampooya/Documents/Apps/FKCloudFunctions/functions" for Cloud Functions...
22:38:54 I functions http function initialized (http://localhost:5001/halogen-basis-356610/us-central1/userIsLegal).
22:47:55 I function[us-central1-userIsLegal] Beginning execution of "userIsLegal"
22:47:55 I function[us-central1-userIsLegal] {
    "verifications": {
        "app": "MISSING",
        "auth": "MISSING"
    },
    "logging.googleapis.com/labels": {
        "firebase-log-type": "callable-request-verification"
    },
    "severity": "INFO",
    "message": "Callable request verification passed"
}
22:47:55 I function[us-central1-userIsLegal] Finished "userIsLegal" in ~1s
```

Apply

Reset

View Logs

So far we have
written and
tested our Cloud
Function let's
deploy it!

$$\sqrt{123}$$

× ÷



STUDY
HARD!



Deploy with one line using CLI
\$ firebase deploy --only functions



When finished you'll see



✓ Deploy complete!



Project Console: <https://console.firebaseio.google.com/project/halogen-basis-356610/overview>





Build ▾

Authentication

App Check

Firestore Database

Realtime Database

Extensions

Storage

Hosting

Functions

Machine Learning

Remote Config

Cloud Messaging

Release & Monitor ▾

Analytics ▾

Engage ▾

All products

Customize your nav!

Blaze

Pay as you go

Modify

Functions

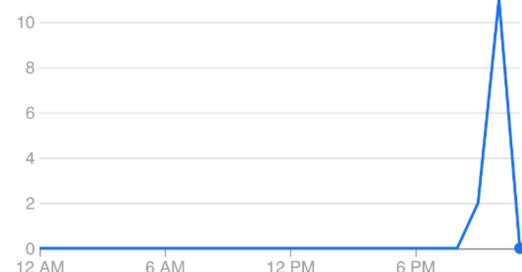
Dashboard

Usage

Check Out Firebase Console to View Your Functions

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
function-zi-trial europe-west1	HTTP Request https://europe-west1-halogen-basis-356610.cloudfunctions.net/function-zi-trial	v1	0	0 / 3000	1m
addTeamName us-central1	HTTP Request https://us-central1-halogen-basis-356610.cloudfunctions.net/addTeamName	v1	0	0 / -	1m
makeShortName us-central1	document.create messages/{documentId}	v1	0	0 / -	1m
userIsLegal us-central1	HTTP Request https://us-central1-halogen-basis-356610.cloudfunctions.net/userIsLegal	v1	13	0 / -	1m

Invocations/second



Additional details

Deployed: 9/25/22, 10:57 PM

Billable instance time: N/A ⓘ

Language: Node.js 16

Memory: 256 MB



Where to Go From Here



Write functions that work with
Firebase triggers such as Cloud
Firestore 



Run functions on a
schedule 



Unit test your functions 



Add security rules to
your Cloud Functions or
use App Check for
protecting your API
resources 



Next time you are in a situation where you
desperately need a backend engineer's help,
just be your own saviour and do it with Cloud
Functions!



$\sqrt{123}$



THANKS!

Do you have any questions?

zamzampooya@gmail.com
[@zamzampooya](https://twitter.com/zamzampooya)
+45 31605586



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

+ x ÷