

Improving Firebase Backend for Flutter App through Cloud Functions



Sasha Denisov, EPAM
Flutter and Dart GDE



About me



SASHA DENISOV



CHIEF SOFTWARE ENGINEER II

Sasha is an experienced software developer with a rich background, in IT for more than 20 years. He worked with different stacks in backend, frontend and mobile fields, but since 2018 his main interest is Flutter.

Sasha is Chief Software Engineer and Head of Flutter Discipline in EPAM. In addition, he is a Flutter and Dart Google Developer Expert and Co-Organizer of Flutter Berlin Meetup community



FlutterHEROES

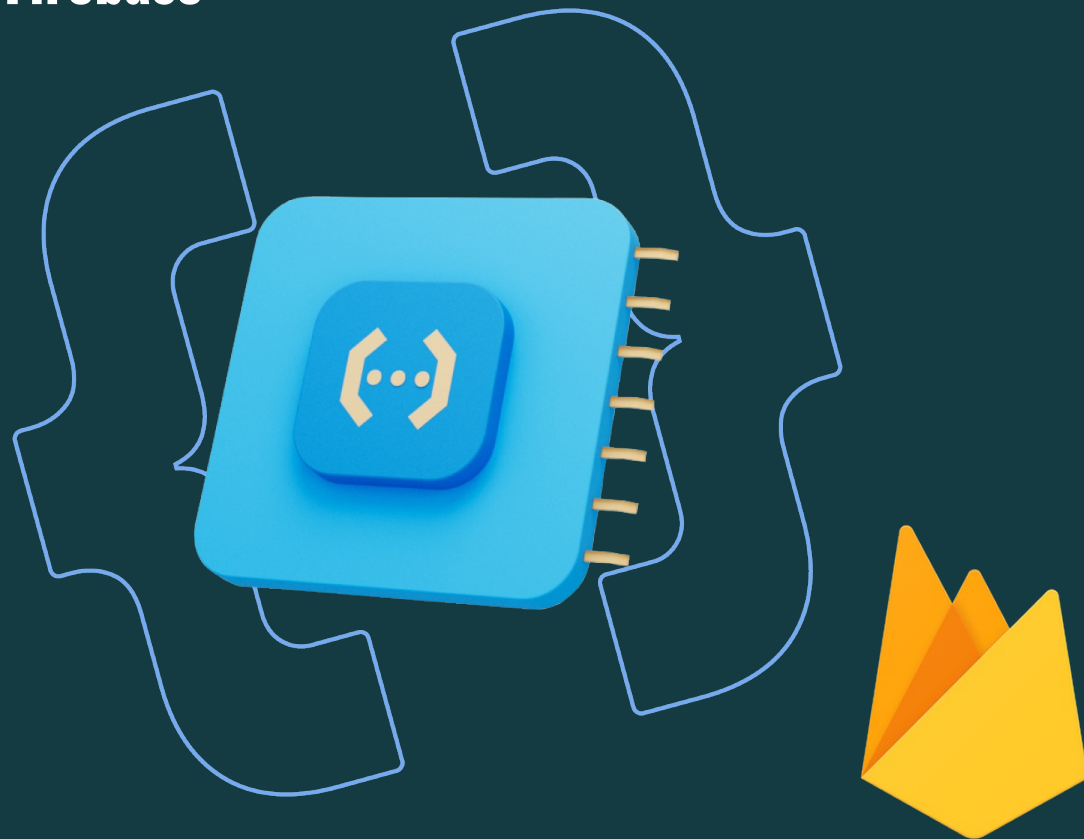


23 February 2024

See you in person or online!

brought to you by /*synesthesia events*

Cloud Functions for Firebase



Clouds


Alibaba

Google

amazon

 HEWLETT
PACKARD

 vmware

ORACLE

 DigitalOcean

 Microsoft

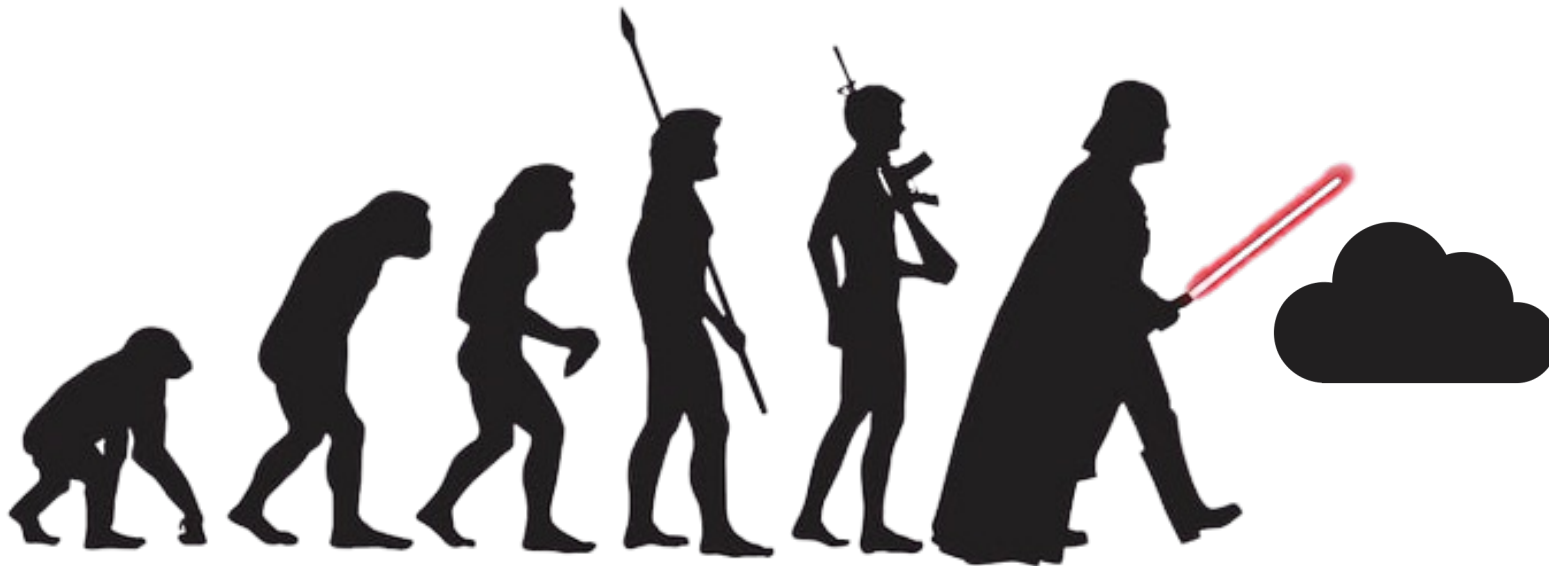
IBM

Cloud Service Characteristics

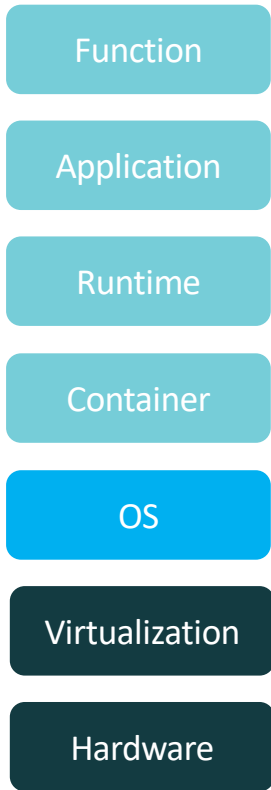
- **On-demand self-service**
- **Broad network access**
- **Resource pooling**
- **Rapid elasticity**
- **Measured service**



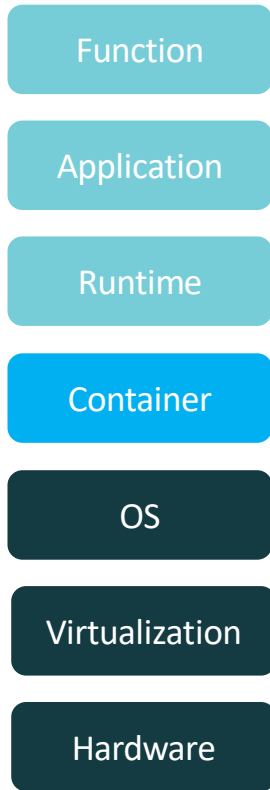
Evolution of Cloud Services



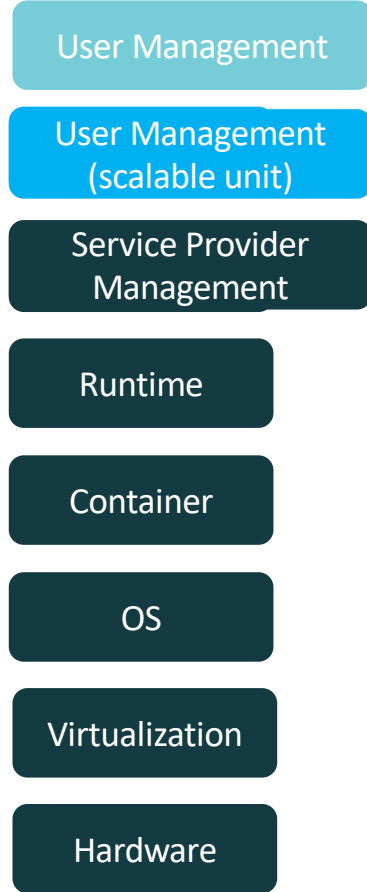
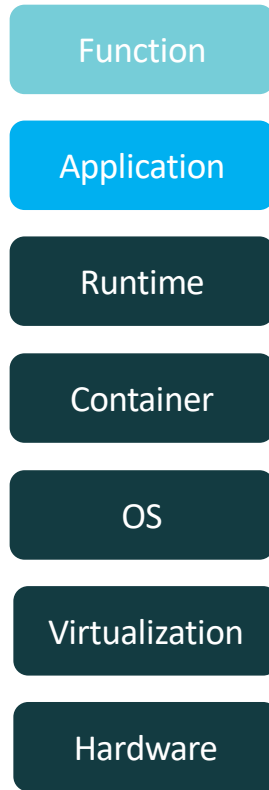
IaaS



CaaS



PaaS



Cloud Functions



The question is

What exactly Firebase backend is?



Firebase

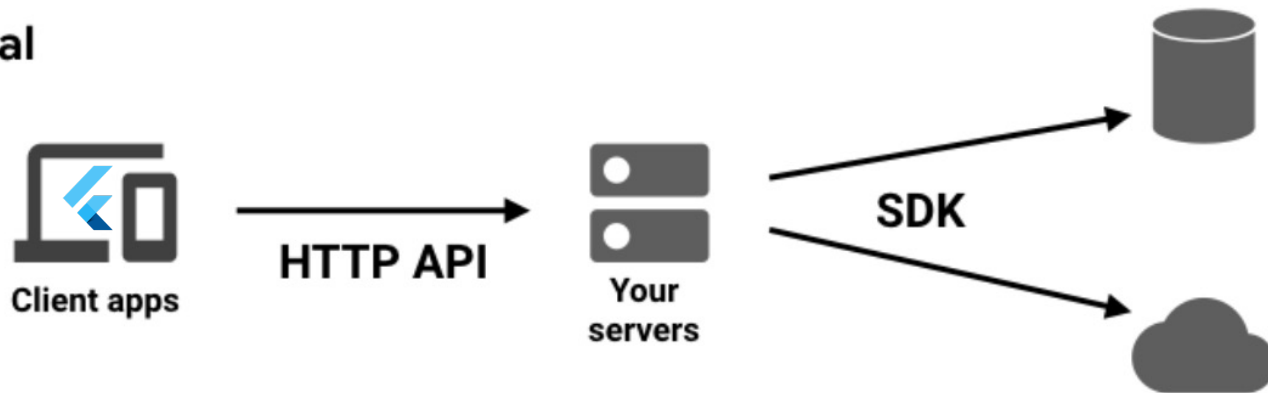


Firebase



Traditional server backend

Traditional



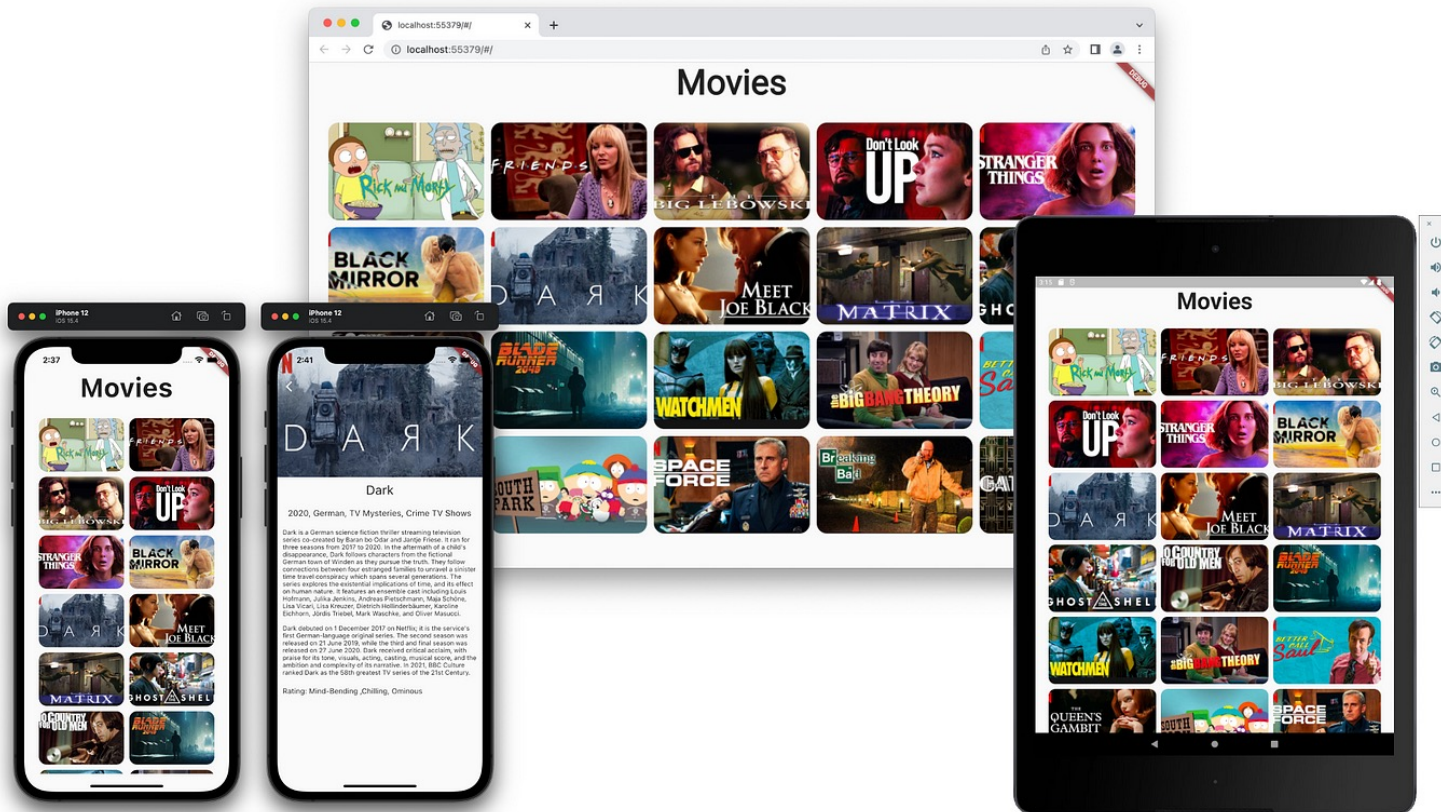
Firestore serverless backend

Firestore



Flutter Application: Movies

https://github.com/denisovav/flutter_tv



Let's add few Firebase Services



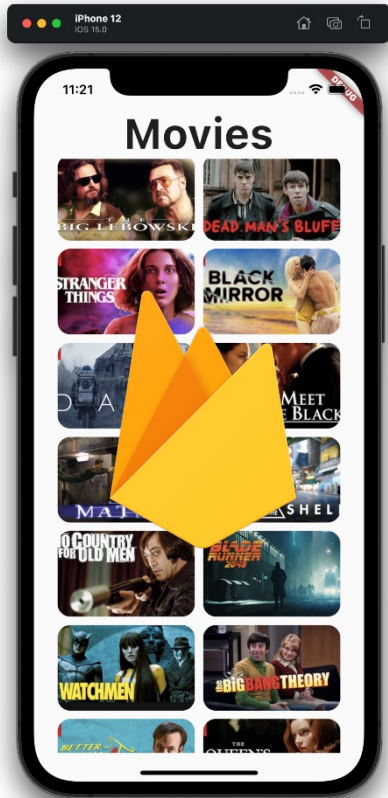
Firestore database for storing movies data



Storage for storing images and videos



Authentication for access control



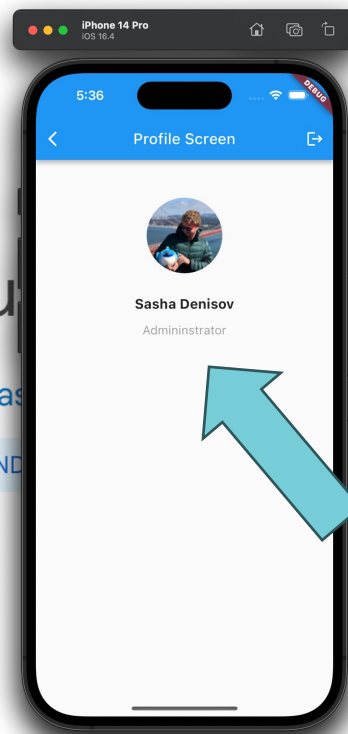
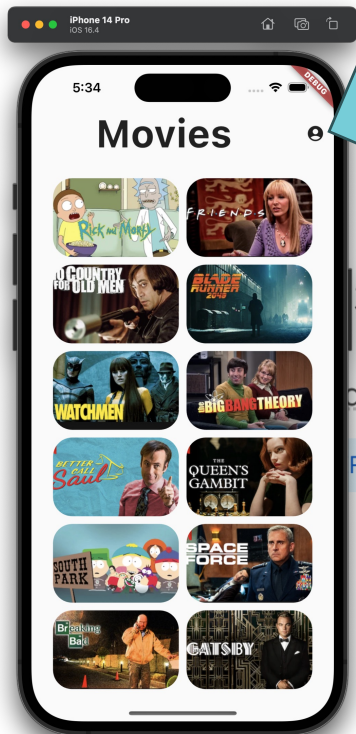
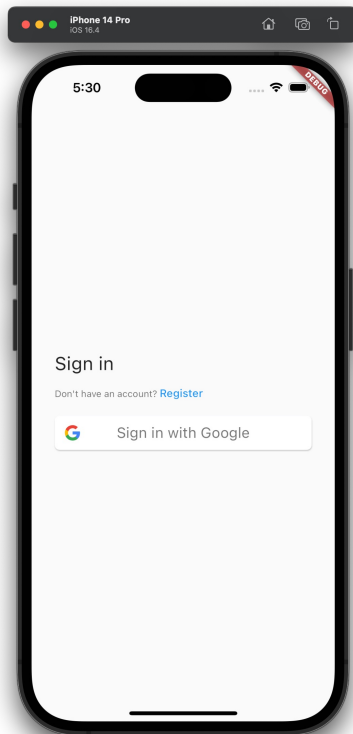
Adding Firebase backend



https://github.com/DenisovAV/flutter_tv/tree/firebase_support

Authentication Added

https://pub.dev/packages/firebase_ui_auth



How to to trigger a function?

Authentication Triggers

1. onCreate
2. onDelete

1st gen

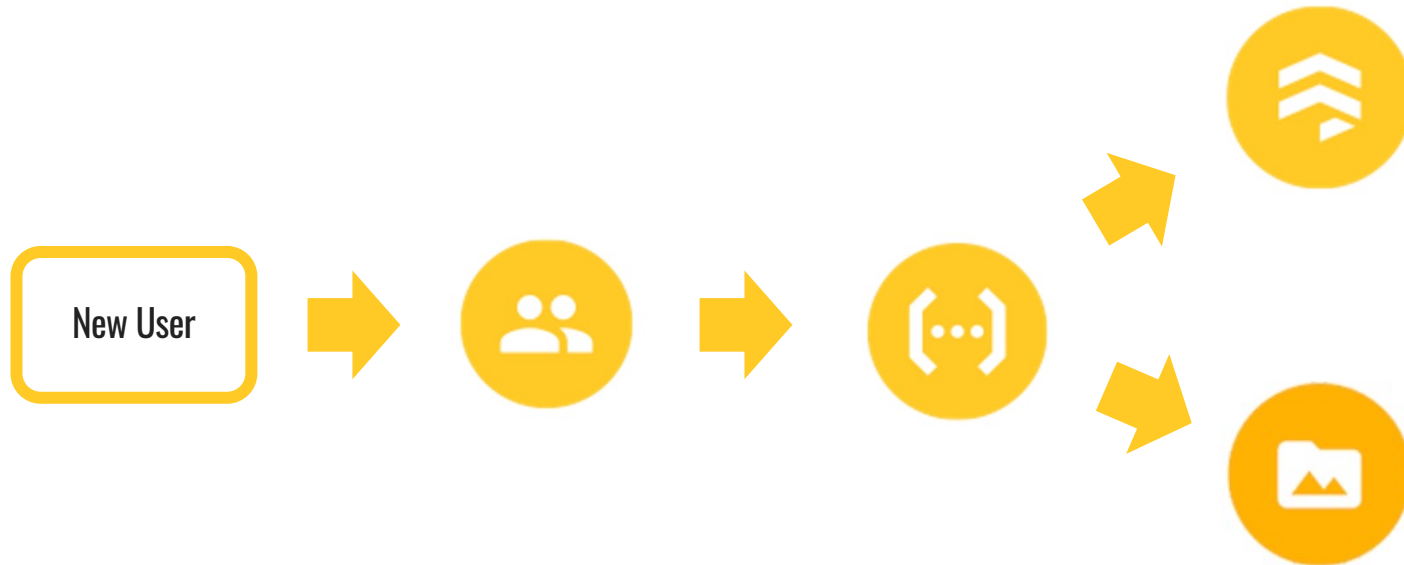


Authentication Functions Using Scenarios

1. Send welcome emails to new users
2. Create a new document in a Firestore collection when a user is created
3. Send a notification to other users when a user is created
4. Delete all user data when a user is deleted



Scenario



How to create a Cloud Function?

1. Go to your project folder

2. Set up Node.js and the Firebase CLI

3. Connect to the project using Firebase CLI

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

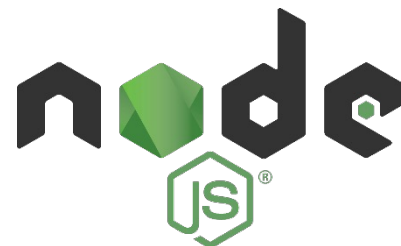
```
firebase login
```

```
firebase init functions
```

4. Write your function code using JS or TS (functions/index.js)

5. Deploy functions to a production environment

```
firebase deploy --only functions
```



Supported Languages



AI ?





Can you create for me Auth onCreate Firebase Cloud Function, which will create a new document in Firestore, move the User Avatar to Cloud Storage, and save its public URL to this document too? And use Typescript. Please :)



Certainly! Here's a Firebase Cloud Function written in TypeScript that creates a new Firestore document, uploads a user avatar to Cloud Storage, and saves the public URL to the Firestore document. Make sure you have Firebase CLI and the necessary Firebase SDKs installed in your project before deploying this function.

```
typescript Copy code

import * as functions from 'firebase-functions';
import * as admin from 'firebase-admin';
import * as path from 'path';
import * as os from 'os';
import * as fs from 'fs';
import * as sharp from 'sharp';

admin.initializeApp();

const firestore = admin.firestore();
const storage = admin.storage().bucket();

// Firebase Auth onCreate function
export const createUserDocument = functions.auth.user().onCreate(async (user) => {
  const userId = user.uid;
  const email = user.email;

  // Create Firestore document
  const userDocRef = firestore.collection('users').doc(userId);

  // Set user data in Firestore
  await userDocRef.set({
    email: email,
    // Add any other user properties here
  });
});
```



```
import * as functions from "firebase-functions";
```

```
export const createUser =  
  exports.createUser = functions.region("europe-west1").auth.user().onCreate(async (user) => {
```

```
    if (user.photoURL) {
```

```
      const avatarFileName = `avatars/${user.uid}.jpg`;
```

```
      const response = await axios.get(user.photoURL, {responseType: "arraybuffer"});  
      const buffer = Buffer.from(response.data, "binary");
```

```
      const file = bucket.file(avatarFileName);  
      await file.save(buffer, {contentType: "image/jpeg", public: true});
```

```
      const uuid = randomUUID();
```

```
      const updatedAvatarURL = `https://firebasestorage.googleapis.com/v0/b/${bucket.name}` +  
        `/o/${encodeURIComponent(file.name)}?alt=media&token=${uuid}`;  
    }
```

```
    // Create a new document in the `users` collection
```

```
    const doc = admin.firestore().collection("users").doc(user.uid);
```

```
    doc.set({  
      user.email, displayName: user.displayName, avatarURL: updatedAvatarURL, admin: false});  
    });
```



5. Deploy functions to a production environment

```
firebase deploy --only functions
```

Functions

Dashboard

Usage




Protect your Functions resources from abuse, such as billing fraud or phishing




[Configure App Check](#)



Looking for logs and health reporting? Visit the Google Cloud Console for a highly customizable [logs view](#), [per-function usage details](#), and [error reporting](#)

[Dismiss](#)

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
onAuthUserCreate europe-west1	 user.create	v1	12	0 / 3000	1m

Items per page: 25  1 - 1 of 1  

(Default) ▾

Panel view

Query builder

🏠 > users > OFFmh1sc6gSR.

☁ More in Google Cloud ▾

📁 movies-cfc57

📁 users

☰ OFFmh1sc6gSRZcmMG0o7GHNRJ2B2

+ Start collection

+ Add document

+ Start collection

movies

OFFmh1sc6gSRZcmMG0o7GHNRJ2B2 >

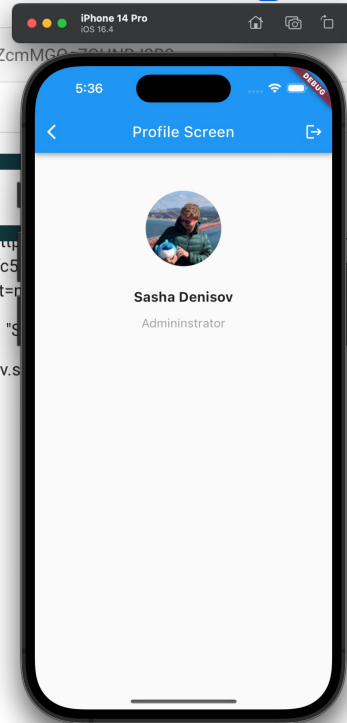
admin: true

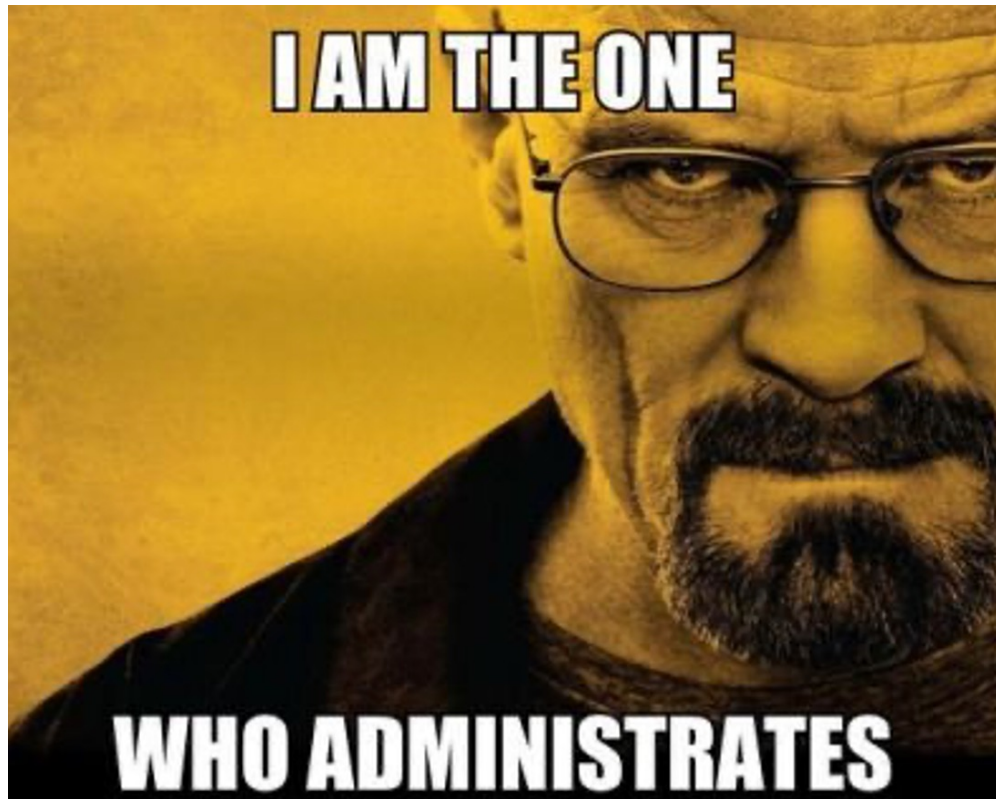
avatarURL: "http://
cfc57...
alt=n...

displayName: "Sasha Denisov"

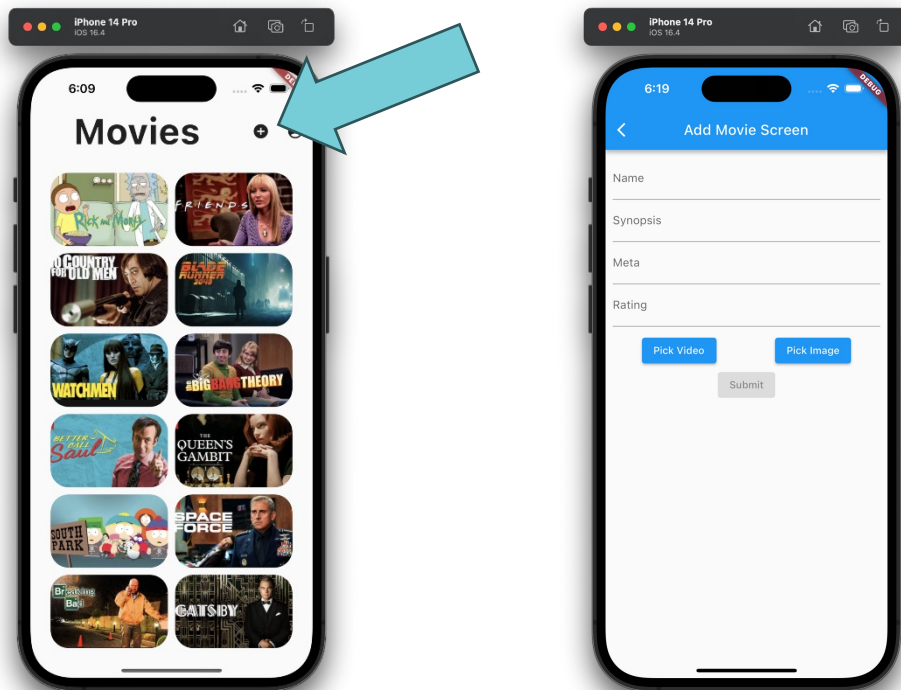
email: "denisov.sasha@gmail.com"

movies-
SRZcmMG0o7GHNRJ2
16b345ba02"





Administrator Capabilities



Storage Triggers

1. **onFinalize**
2. **onDelete**
3. **onArchive**
4. **onMetadataUpdate**

1. **onObjectFinalized**
2. **onObjectDeleted**
3. **onObjectArchived**
4. **onMetadataUpdated**

1st gen

2nd gen



Cloud Functions Generation 2

Feature	Cloud Functions (1st gen)	Cloud Functions (2nd gen)
Image registry	Container Registry or Artifact Registry	Artifact Registry only
Request timeout	Up to 9 minutes	<ul style="list-style-type: none">• Up to 60 minutes for HTTP-triggered functions• Up to 9 minutes for event-triggered functions
Instance size	Up to 8GB RAM with 2 vCPU	Up to 16GiB RAM with 4 vCPU
Concurrency	1 concurrent request per function instance	Up to 1000 concurrent requests per function instance
Traffic splitting	No supported	Supported

Supported Languages in Gen2





Google
Cloud Run

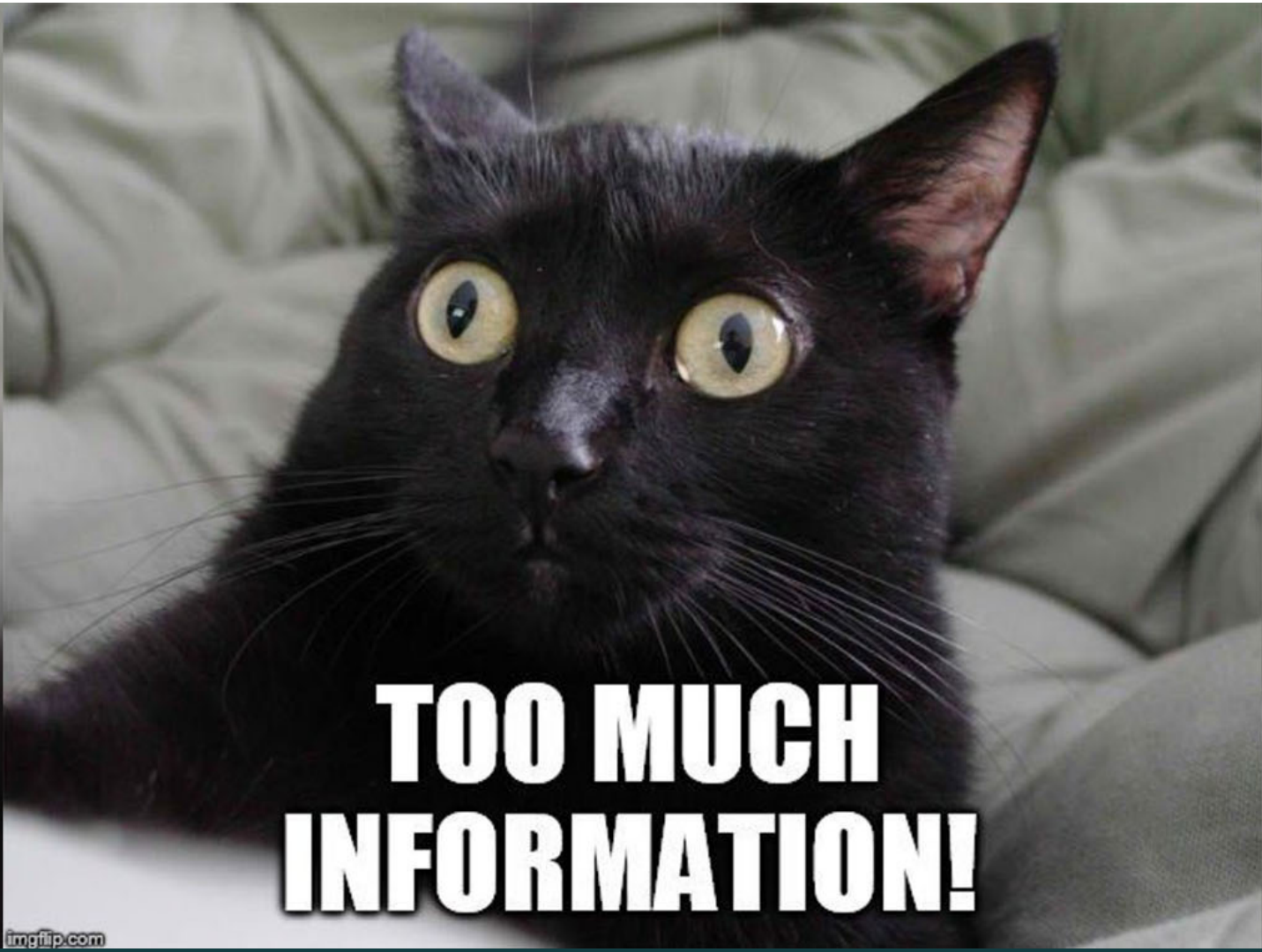
Dart on Server Side



Serverpod



Dart Frog



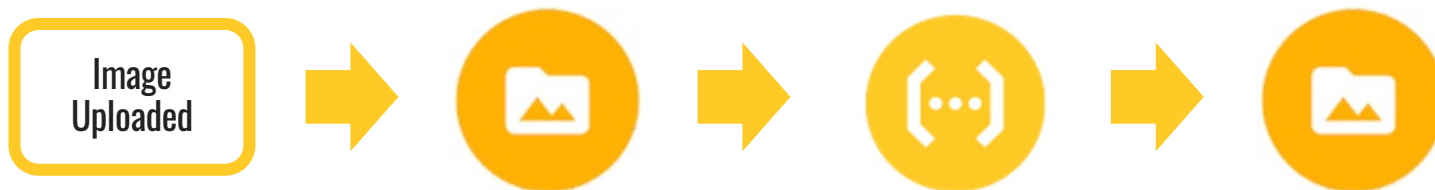
**TOO MUCH
INFORMATION!**

Storage Functions Using Scenarios

1. Image Processing
2. Notifications
3. Data validation and Sanitization
4. File Metadata Management
5. Backup and Archiving
6. Access Control
7. Audio and Video Processing
8. Integration with Third-party Services
9. Analytics
10. Cleanup Operations



Scenario




```
import * as functions from "firebase-functions/v2";
```

```
export const compressImage =  
exports.compressMedia = functions.storage.onObjectFinalized({cpu: 2}, async (event) => {  
  const fileBucket = event.data.bucket; // Storage bucket containing the file.
```

```
  if (!contentType.startsWith("image/")) {  
    return console.error("This is not an uploaded image.");  
  }
```

```
  const bucket = admin.storage().bucket(fileBucket);  
  const downloadResponse = await bucket.file(filePath).download();  
  const imageBuffer = downloadResponse[0];
```

```
  const meta = await sharp(imageBuffer).metadata();
```

```
  const rotatedBuffer = (meta.width! < meta.height! ? await sharp(imageBuffer).rotate(90).toBuffer() : imageBuffer);
```

```
  const processedBuffer = await sharp(rotatedBuffer).resize({width: 600, height: 300, withoutEnlargement: true,  
  }).toBuffer();
```

```
  // Upload the processed image.
```

```
  const metadata = {contentType: contentType};  
  const file = bucket.file(filePath);  
  await file.save(processedBuffer, {  
    metadata: metadata,  
  });
```

```
  return console.log("Image uploaded!");  
});
```



Functions

Dashboard

Usage





Protect your Functions resources from abuse, such as billing fraud or phishing

[Configure App Check](#)



Looking for logs and health reporting? Visit the Google Cloud Console for a highly customizable [logs view](#), [per-function usage details](#), and [error reporting](#)

[Dismiss](#)

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
compressImage europe-west1	 <code>object.finalized</code> Bucket: movies-cfc57.appspot.com	v2	1	0 / 100	1m
onAuthUserCreate europe-west1	 <code>user.create</code>	v1	9	0 / 3000	1m

Items per page: 25 ▾ 1 – 2 of 2 < >

The question is

How to notify users about new content?



Firestore Triggers

1. `onCreate`
2. `onUpdate`
3. `onDelete`
4. `onWrite`

1st gen

1. `onDocumentCreated`
2. `onDocumentUpdated`
3. `onDocumentDeleted`
4. `onDocumentWritten`

2nd gen

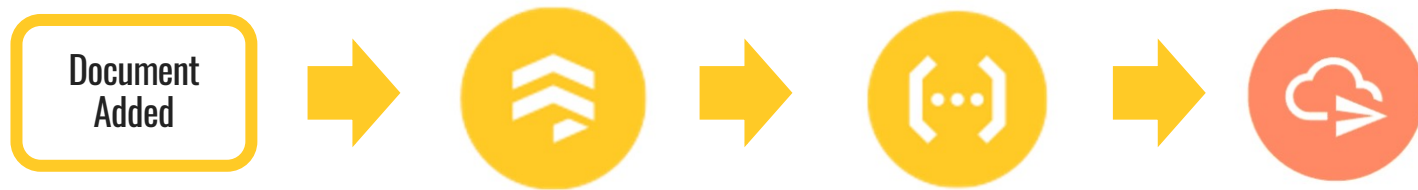


Firestore Functions Using Scenarios

1. Data validation and Sanitization
2. Notifications
3. Data Aggregation
4. Integration with External Services
5. Automatic Data Management
6. Access Control
7. Chaining Database Operations
8. Complex Business Logic
9. Analytics
10. SEO Operations



Scenario



```
import * as functions from "firebase-functions/v2";
```

```
functions.setGlobalOptions({  
  region: "europe-west1",  
});
```

```
export const sendPushNotification = functions.firestore.onDocumentCreated("movies/{id}", async (event) => {
```

```
  const name = movie.get("name");  
  const image = movie.get("image");
```

```
  const notification: admin.messaging.Message = {notification: {title: `${name} added!`,  
    body: "New movie was added to movies catalog, please enjoy!", imageUrl: image, },  
    topic: "movies",  
  };
```

```
  try {  
    const response = await admin.messaging().send(notification);  
    console.log("Successfully sent message:", response);  
  } catch (error) {  
    console.error("Error sending message:", error);  
  }  
});
```

The TypeScript logo, consisting of a blue square with the white letters "TS" inside.

Functions

Dashboard

Usage






Protect your Functions resources from abuse, such as billing fraud or phishing

[Configure App Check](#)



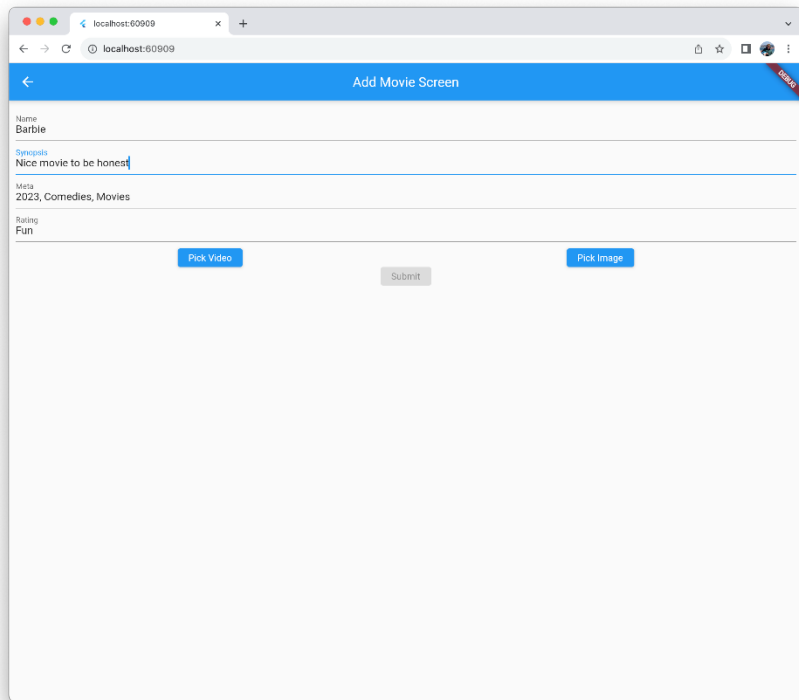
Looking for logs and health reporting? Visit the Google Cloud Console for a highly customizable [logs view](#), [per-function usage details](#), and [error reporting](#)

[Dismiss](#)

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
compressImage europe-west1	 <code>object.finalized</code> Bucket: movies-cfc57.appspot.com	v2	2	0 / 100	1m
sendPushNotification europe-west1	 <code>document.created</code>	v2	4	0 / 100	1m
onAuthUserCreate europe-west1	 <code>user.create</code>	v1	7	0 / 3000	1m

Items per page: 25 ▼ 1 - 3 of 3 < >

Firestore Cloud Messaging



Rest API



```
export const getMoviesData = function https onRequest(async (req, res) => {
  const snapshot = await admin.firestore().collection('movies').get();

  if (snapshot.empty) {
    res.status(400).send({error: "Collection of movies is empty."});
    return;
  }

  const data: any[] = [];

  snapshot.forEach((doc) => {
    data.push({
      id: doc.id,
      ...doc.data(),
    });
  });

  res.status(200).send({data: data});
});
```



Functions

Dashboard

Usage



Protect your Functions resources from abuse, such as billing fraud or phishing

[Configure App Check](#)



Looking for logs and health reporting? Visit the Google Cloud Console for a highly customizable [logs view](#), [per-function usage details](#), and [error reporting](#)

[Dismiss](#)

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
compressImage europe-west1	<code>object.finalized</code> Bucket: movies-cfc57.appspot.com	v2	3	0 / 100	1m
getMoviesData europe-west1	Request <code>https://getmoviesdata-ujjihq4gfa-ew.a.run.app</code>	v2	3	0 / 100	1m
sendPushNotification europe-west1	<code>document.created</code>	v2	5	0 / 100	1m
onAuthUserCreate europe-west1	<code>user.create</code>	v1	8	0 / 3000	1m

Items per page: 25 1 - 4 of 4

pretty print

```
{
  "data": [
    {
      "id": "rick",
      "trailer": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/videos%2Frick.mp4?alt=media&token=194fefal-5ef7-4af4-aede-a80621068571",
      "image": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/images%2Frick.png?alt=media&token=c78c4bd6-8d17-450d-beaa-615f64fd8c3a",
      "meta": "2021, Sci-Fi TV, Sitcoms, Adult, Animation",
      "name": "Rick and Morty",
      "rating": "Absurd, Quirky, Irreverent",
      "synopsis": "Rick and Morty is an American adult animated science fiction sitcom created by Justin Roiland and Dan Harmon for Cartoon Network's nighttime progr",
    },
    {
      "id": "friends",
      "trailer": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/videos%2Ffriends.mp4?alt=media&token=fb656cec-e5a5-41fe-b84c-7d4884a12c1a",
      "image": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/images%2Ffriends.png?alt=media&token=1c2ee9ca-ad02-443c-aff2-9aa3e2fc1077",
      "meta": "2003, Sitcoms, TV Comedies, US TV Shows",
      "name": "Friends",
      "rating": "Witty, Quirky",
      "synopsis": "Friends is an American television sitcom created by David Crane and Marta Kauffman, which aired on NBC from September 22, 1994, to May 6, 2004, la",
    },
    {
      "id": "country",
      "trailer": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/videos%2Fcountry.mp4?alt=media&token=f6f07890-e929-49e8-84f2-79bb7156ad07",
      "image": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/images%2Fcountry.png?alt=media&token=2f5f93ea-4218-4842-a854-9b20560caed1",
      "meta": "2007, Movies Based on Books, Crime Movies, Dramas",
      "name": "No Country for Old Men",
      "rating": "Violent, Gritty, Dark",
      "synopsis": "No Country for Old Men is a 2007 American neo-Western crime thriller film written and directed by Joel and Ethan Coen, based on Cormac McCarthy's ",
    },
    {
      "id": "bladerunner",
      "trailer": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/videos%2Fbladerunner.mp4?alt=media&token=d3fd3673-7657-4366-b76f-c4b981c512e",
      "image": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/images%2Fbladerunner.png?alt=media&token=d1e94990-e346-4773-b404-2c2ae8dff3f0",
      "meta": "2017, Sci-Fi Movies, Action & Adventure, Crime Movies",
      "name": "Blade Runner 2049",
      "rating": "Dark",
      "synopsis": "Blade Runner 2049 is a 2017 American neo-noir science fiction film directed by Denis Villeneuve and written by Hampton Fancher and Michael Green. i",
    },
    {
      "id": "watchmen",
      "trailer": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/videos%2Fwatchmen.mp4?alt=media&token=ad7194ac-857d-4ef7-95c7-11e6569132ac",
      "image": "https://firebasestorage.googleapis.com/v0/b/movies-cfc57.appspot.com/o/images%2Fwatchmen.png?alt=media&token=2ef6b330-3d30-4610-8d35-312444045d30",
    }
  ]
}
```

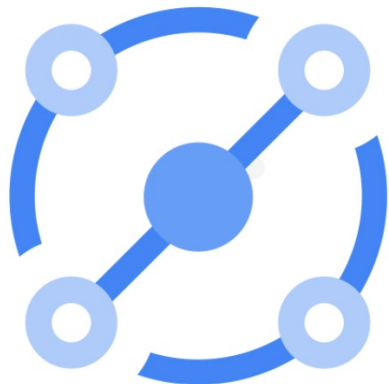
Callable

await FirebaseFunctions.*instance*.httpsCallable('yourFunctionName').call();

More Triggers

- 1. Firebase Alerts Triggers**
- 2. Blocking Auth Triggers**
- 3. Analytics Triggers**
- 4. Remote Config Triggers**
- 5. Test Lab Triggers**
- 6. Pub/Sub Triggers**

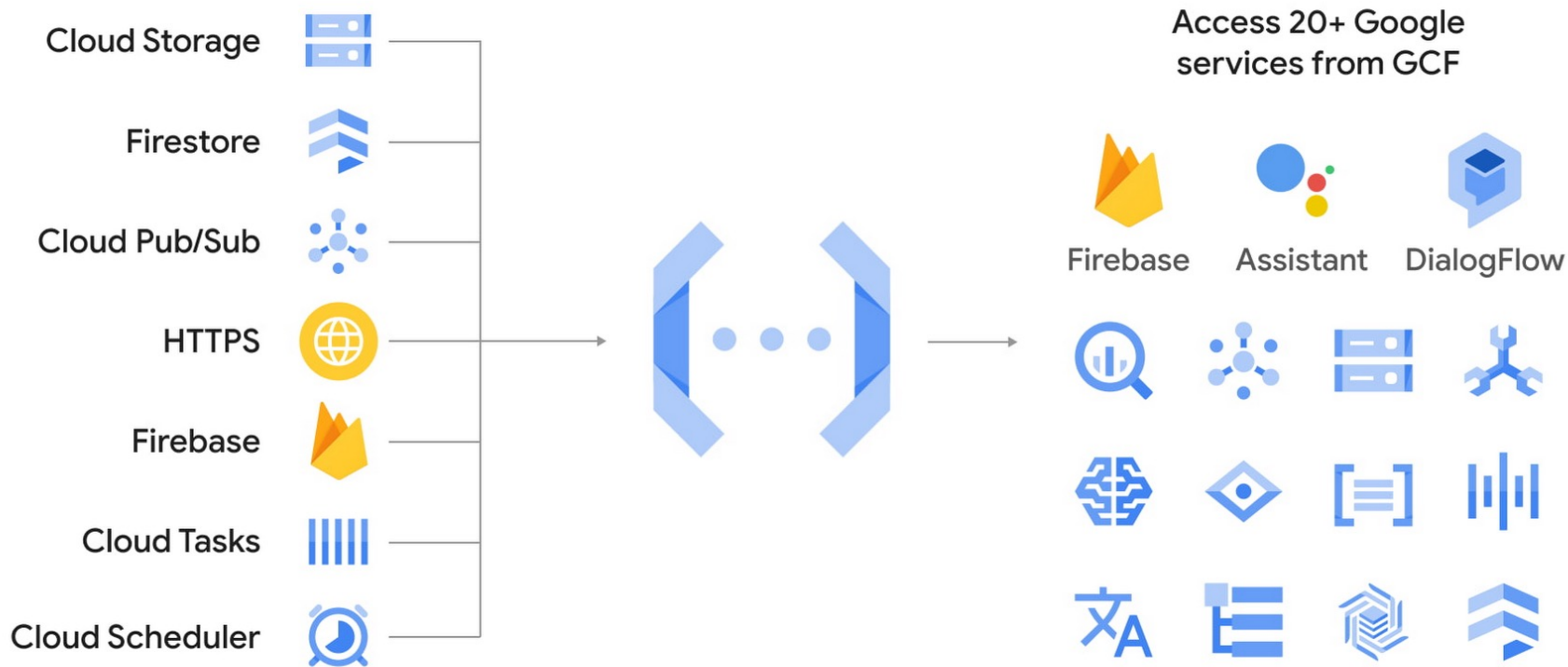
Custom Events



Eventarc



Cloud Functions: the glue that connects cloud services



Google Cloud Platform



Google Cloud

Movies

Search (/) for resources, docs, products, and more Search

Cloud Functions

Functions

[+ CREATE FUNCTION](#) [REFRESH](#)

Filter Filter functions

<input type="checkbox"/>	Environment	Name ↑	Last deployed	Region	Recommendation	Trigger	Runtime	Memory allocated	Executed function	Actions
<input type="checkbox"/>	✓ 2nd gen	compressImage	Sep 24, 2023, 6:18:47 PM	europa-west1		Bucket: movies-cfc57.appspot.com	Node.js 16	256 MiB	compressImage	⋮
<input type="checkbox"/>	✓ 2nd gen	getMoviesData	Sep 24, 2023, 6:18:45 PM	europa-west1		HTTP	Node.js 16	256 MiB	getMoviesData	⋮
<input type="checkbox"/>	✓ 1st gen	onAuthUserCreate	Sep 24, 2023, 6:18:44 PM	europa-west1		Firebase Authentication (Preview)	Node.js 16	256 MB	onAuthUserCreate	⋮
<input type="checkbox"/>	✓ 2nd gen	sendPushNotification	Sep 24, 2023, 6:19:09 PM	europa-west1		Cloud Firestore	Node.js 16	256 MiB	sendPushNotification	⋮

Google Cloud Platform

Google Cloud ▼ Movies ▼

Search (/) for resources, docs, products, and more 🔍 Search

Cloud Functions ← Create function

Configuration — **2** Code

Runtime ?

- Go 1.16
- Go 1.13
- Java 17
- Java 11
- Node.js 20**
- Node.js 18
- Node.js 16
- Node.js 14

Entry point * ? TEST FUNCTION

helloHttp

Press Option+F1 for Accessibility Options.

```
1  const functions = require('@google-cloud/functions-framework');
2
3  functions.http('helloHttp', (req, res) => {
4    | res.send('Hello ${req.query.name || req.body.name || 'World'}!');
5  });
6
```

**I USED TO BE A
BACKEND JAVA DEVELOPER**


SO LET'S GO BACK TO THE OLD DAYS

imgflip.com

Runtime
Java 11

Entry point *
com.example.Example

TEST FUNCTION

 The specified entry point might not be present in your source code. Please ensure the entry point in your code matches the input field. [DISMISS](#)

Source code
Inline Editor

- src
 - main
 - java
 - com
 - example
 - Example.java

- pom.xml


Press Option+F1 for Accessibility Options.

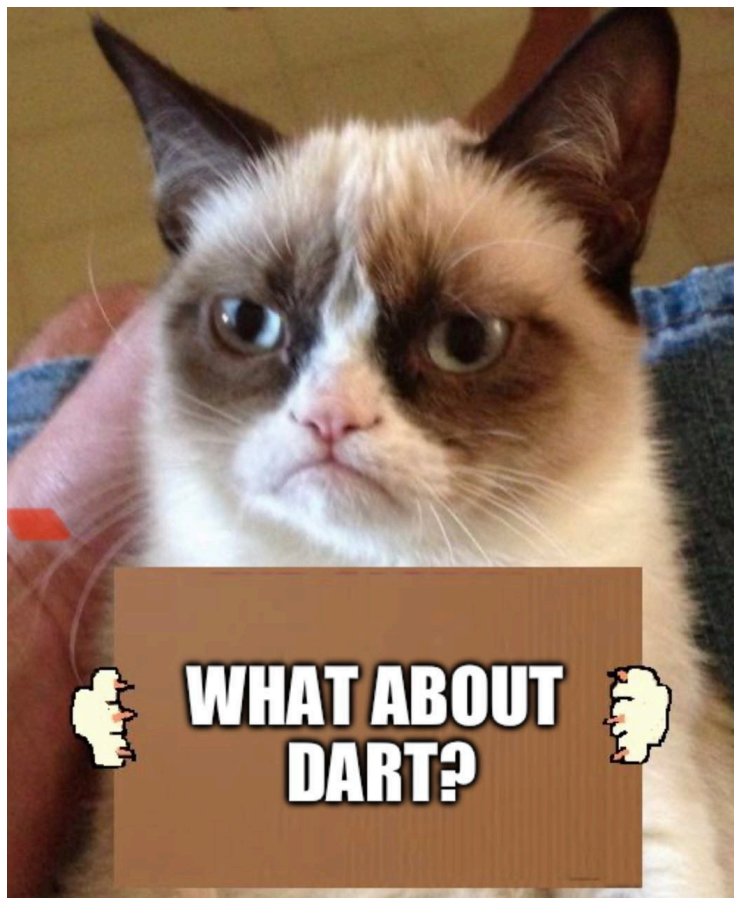
```

1 package gcfv2;
2
3 import java.io.BufferedWriter;
4
5 import com.google.cloud.functions.HttpFunction;
6 import com.google.cloud.functions.HttpRequest;
7 import com.google.cloud.functions.HttpResponse;
8 import com.google.firebase.cloud.FirestoreClient;
9 import com.google.cloud.firestore.Firestore;
10 import com.google.cloud.firestore.QuerySnapshot;
11 import com.google.cloud.firestore.DocumentSnapshot;
12 import java.io.BufferedWriter;
13 import java.io.IOException;
14 import java.util.HashMap;
15 import java.util.Map;
16
17
18 public class GetMovies implements HttpFunction {
19     public void service(final HttpRequest request, final HttpResponse response) throws Exception {
20         Firestore firestore = FirestoreClient.getFirestore();
21
22         QuerySnapshot snapshot;
23         try {
24             snapshot = firestore.collection("movies").get().get();
25         } catch (Exception e) {
26             BufferedWriter writer = response.getWriter();
27             response.setStatuscode(500);
28             writer.write("Error fetching data: " + e.getMessage());
29             return;
30         }
31
32         if (snapshot.isEmpty()) {
33             BufferedWriter writer = response.getWriter();
34             response.setStatuscode(400);
35             writer.write("Collection of movies is empty.");
36         }
37     }
38 }
    
```

PREVIOUS **DEPLOY** CANCEL

Supported Languages

- Node.js (12, 14, 16, 18, and 20)
- Python (3.7, 3.8, 3.9, 3.10 and 3.11)
- Go (1.13, 1.16, 1.18, 1.19, 1.20 and 1.21)
- Java (11 and 17)
- .NET (6.0 and Core 3.1)
- Ruby (2.6, 2.7, 3.0 and 3.2)
- PHP (7.4, 8.1 and 8.2)



Using Dart to write Firebase Functions? #383



Closed

BenHayat opened this issue on Jan 15, 2019 · 11 comments



BenHayat commented on Jan 15, 2019



Hello Team;

When is Firebase going to support Dart to write Cloud Functions? We're planning to use ONE language to build the frontend (Flutter) and Backend (FB Functions) than going with Node JS to write backend.

Thanks!



44

...



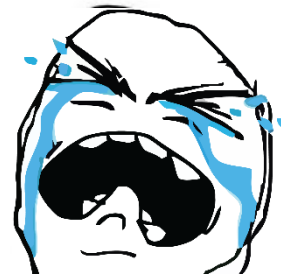
katowulf commented on Apr 16, 2020



Current status: We don't have any plans to add support for languages other than node in the near future. We'll update if that changes. Locking thread to prevent additional ETA requests.



firebase locked as resolved and limited conversation to collaborators on Apr 16, 2020





Google
Cloud Run

functions_framework 0.4.3+1

Published 43 days ago • cloud.google.com [Dart 3 compatible](#)

[SDK](#) [DART](#) [FLUTTER](#) [PLATFORM](#) [ANDROID](#) [IOS](#) [LINUX](#) [MACOS](#) [WINDOWS](#)

 141

[Readme](#) [Changelog](#) [Installing](#) [Versions](#) [Scores](#)

Functions Framework for Dart

This is a **community-supported project**, meaning there is no official level of support. The code is not covered by any SLA or deprecation policy.

Feel free to start a [discussion](#) to share thoughts or open [issues](#) for bugs and feature requests.

Functions Framework Unit Tests Lint Test Conformance Tests

Dart

 Dart Unit CI **passing**

 Dart Lint CI **passing**

 Dart Conformance CI **passing**

The future is bright (especially for Flutter developers)



THE FINAL SLIDE



https://github.com/DenisovAV/flutter_tv/tree/firebase_support



[@shuregdenisov](https://twitter.com/shuregdenisov)



THANK YOU!
QUESTIONS?