Getting Started with Firebase: Cloud Functions

Introduction

Firebase Cloud Functions provide a serverless environment to run backend code in response to events triggered by Firebase features and HTTPS requests. This is particularly useful for applications requiring background processing like sending notifications, performing complex calculations, or integrating with third-party services. In this tutorial, we'll learn how to set up and use a simple Firebase Cloud Function in a web application.

Prerequisites

Before we get started, make sure you have the following:

- A Google account (for Firebase authentication)
- Node.js and Angular installed on your machine

Setting Up Firebase Project

1. Create a Firebase Project:

- Go to the Firebase Console (https://console.firebase.google.com/).
- Click on "Add project" and follow the instructions to create a new project.

2. Add Firebase to Your Web App:

- Inside your project, click on "Add app" and choose the web icon.
- Register your app and note down the configuration details.

3. Enable Cloud Functions:

- In the Firebase Console, navigate to the "Functions" section.
- Click on "Get started" to set up Cloud Functions in your project.

Creating a Cloud Function

Let's create a simple Cloud Function that calculates the sum of an array of numbers.

Define the Function:

Inside the functions/index.js file, write a function to handle HTTP requests.

Example:

```
const functions = require('firebase-functions');

// Define the Cloud Function
exports.computeSum = functions.https.onRequest((request, response) => {
   // Ensure request body is not empty
   if (!request.body || !request.body.numbers) {
     return response.status(400)
      .send('Input must contain an array of numbers');
   }

   // Extract the array of numbers from the request body
   const numbers = request.body.numbers;

   // Check if the input is an array
   if (!Array.isArray(numbers)) {
     return response.status(400)
      .send('Input must be an array of numbers');
   }

   // Use the reduce function to calculate the sum
   const sum = numbers.reduce((accumulator, currentValue)
      => accumulator + currentValue, 0);

   // Send the computed sum in the response
   response.status(200).json({ sum });
});
```

Deploy the Function:

Deploy your function using: *firebase deploy --only functions*.

Testing the Cloud Function

Access the Function:

After deployment, you'll receive a URL for your function.
Access this URL in your browser or use a tool like Postman to send a request:

POST: https://URL-to-function/computeSum

With a similar JSON:

```
{"numbers": [800, 100, 800, 800, 0, 529.99, 859.98, 800, 999, 12.32]}
```

You should get a response similar to this one:

```
{"sum":5701.289999999999}
```

Integrating Cloud Function with Your Web App

Use the provided URL to make requests to your Cloud Function.
Example in TypeScript:

```
calculateTotalFirebase(): void {
 // Extract the numbers from your orders

 const numbers = this.orders.map(order => order.price);
      // Make the HTTP request to your Cloud Function
      this.http.post<{sum:number}>
      ('https://URL-to-function/computeSum',
      {"numbers": numbers })
            .subscribe(result => {
            // Update the total
            this.total = result.sum;
      }, error => {
            console.error(error);
   });
```

Conclusion

Congratulations! You've just created and deployed a Firebase Cloud Function and integrated it with a web application. Now you can handle server-side logic without the need for a dedicated server. Happy coding!

Public Git Repository

Access the complete code for this tutorial on GitHub: Firebase Cloud Functions Demo. Feel free to fork and adapt it as needed.

Feel free to fork the repository and customise it according to your needs.

Frontend URL:
https://github.com/AndreiFoidas/soa-fe
Backend URL:
https://github.com/AndreiFoidas/soa-be