Google Firebase Services

Research document

The research was done by
a student Vitali Bestolkau(4138872)
from IPS-DB03 group

Table of Contents

Version history

Version	Date	Change
1.0	13-01-2022	Created file and added main content
1.1	17-01-2022	Benefits and final Conclusion were added

Introduction

During the development of every application at some point developers begin considering implementing some security logic to ensure that the application will not be vulnerable to people that intend to gain certain privileges to become in charge of the application that may drastically affect the whole system depending on the person's intentions. I found myself in the same situation. That's when I was suggested to use Firebase Authentication service.

However, Firebase can help developers with a range of issues those may encounter. And later on I myself started using a number of the services provided by Firebase that work perfectly fine and appeared to be more than enough even with free of charge versions of them. Agree, that is way better than some Azure that is trying to charge you for every step you make. Nevertheless, there we are going to talk about Firebase.

Firebase

To begin with, what is Firebase? Firebase is a platform developed by Google that helps you quickly develop high-quality apps and grow your business. Sounds pretty cool, doesn't it? But let's look a bit more in the depth and check the services that Firebase provides us with:

- Realtime Database: We have by far now known about the very first product of Firebase, and this product is the reason behind this company's success.

- Authentication: This is another application of Firebase of which we have already talked about. The authentication would take place through Email and Password, mobile number and Social media platforms. Firebase provides the integration of all these types of authentication methods.

- Firebase Cloud Messaging: This type of messaging platform allows the developer to build an interface that would provide a reliable connection between the server and devices so that messages and notifications are received at no extra cost.

- Hosting: Hosting service gives you an opportunity to deploy a single-page web app, a mobile app landing page, or a progressive web app without all the hassle.

- Cloud Storage: Cloud Storage is designed to help you quickly and easily store and serve user-generated content, such as photos and videos. As well as you can store there images essential for your app and to access them right from the code

And way more. This is only the tip of the iceberg when we talk about the full potential of Firebase. Moreover, doesn't Firebase allow you to use as many services but it also makes it easy to implement them with the help of the official Firebase documentation, containing code samples for different languages besides their clear explanation of how the services work.

As you may have already realised, Firebase provides a great number of services and that makes it very difficult to talk about Firebase in whole. That's why there I will describe three main features that I was using while developing my own app: Firebase Authentication, Firebase Hosting and Firebase Cloud Storage.

Firebase Authentication

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

If you ever considered using some kind authentication system for your application you should already be acquainted with the term JWT. But in order to make it clear for everyone let's quickly go through this term. JWT, or JSON Web Token, is an open standard used to share security information between two parties — a client and a server. Each JWT contains encoded JSON objects, including a set of claims (I will mention that later). JWTs are signed using a cryptographic algorithm to ensure that the claims cannot be altered after the token is issued. So, JWT helps transfer security-related data from client to server or vice versa. However, if you decide to implement it yourself without any prior experience, you will see a lot of boilerplate that needs to be implemented, which is usually not so fancy. And there Firebase Authentication comes.

Firebase describes in its documents the way to use its authentication. In case you decide to keep it very simple, you won't even need a great understanding of JWT logic. You can simply sign users in with email and password and be happy.

However, imagine you need to access some more user-specific info than email and password. Then it becomes a little bit more tricky. For example, I had to check on the client side, if the user is an admin or not. For this reason, I had to pass custom claims from my server and I also decided then to create custom tokens (aka JWTs) on the server side as Firebase provides this opportunity as well. In order to be able to do that in your server you will need to configure a connection to the Firebase with an Admin SDK. After sending a newly created token on your server, send it to the client part and sign in with a custom token (method, provided by Firebase as well). If the sign in was successful, then you can access the claims by decoding the token.

Additionally, there is one more great feature to mention. You noticed that I mentioned “decoding a JWT/token”. Maybe it means that we can also encode them? And may the users just encode your token, set in the token, for example, an admin claim to “true” and put it into the application in order to use all the admin's features inside the app? Yes, they can. But for that reason you can make use of one more feature, provided by Admin SDK, and that's token verification. As I have already mentioned, all JWT are signed and the ones we use are signed by Firebase specifically. Whilst, when somebody decides to change some claim inside a token by themselves, then that token won't be signed by Firebase, as it wouldn't be a token provided by their service. And token verification simply checks if the token was signed by Firebase or not. That way on your server part you will be able to check the token that you get and send a response back to the client.

So let's outline the main benefits for the Firebase Authentication.

Benefits

- Firebase significantly reduces boilerplate compared to standard JWT implementation, not including other features.

- Level of security provided by Firebase is one of the greatest

- Supports a number of main programming languages such as Java, Node.js, C# and so forth.

- Easy to implement and Firebase has excellent documentation explaining all the steps you need to take.

- A not so experienced developer can keep everything simple and still working, but at the same time Firebase provides features to get more control of your Authentication process, such as Admin SDK.

Firebase Hosting

Firebase Hosting is production-grade web content hosting for developers. With a single command, you can quickly deploy web apps and serve both static and dynamic content to a global CDN (content delivery network).

This service is quite a straightforward one. It helps developers to deploy their websites so that it becomes available for the public. Firebase makes it easy to configure the deployment for your web application, by downloading firebase to your application directly using either npm or yarn and following all the steps in the console. Then firebase-related files that contain the deployment configuration will be added to your application. Also, Firebase can add additional Workflows to your GitHub repository, which will check your code and deploy it to the server, in case all the checks were passed successfully. That way your CI/CD pipeline will be almost completely done within several minutes thanks to Firebase Hosting service.

There are the main advantages of using exactly Firebase Hosting:

Benefits

- Extremely easy to implement using npm or yarn.

- Firebase does all the job for you, including Workflows in GitHub and creating additional files in your application.

- Fastly and securely delivers the app while deployment to the server.

Firebase Cloud Storage

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality.

Cloud Storage service is a greater helper for those who need to store a lot of media files and use them inside their application. It was always a problem to store this kind of files using simple SQL/MySQL/etc. databases. With the Firebase Cloud Storage there is no more need to worry about these issues. As it was mentioned in the description, this service allows developers to effectively and securely download, store, upload and manage photos, videos of different network quality.

Also, it's good to mention that Firebase SDKs for Cloud Storage perform uploads and downloads regardless of network quality. Uploads and downloads are robust, meaning they restart where they stopped, saving your users time and bandwidth.

The main benefits of Firebase Cloud Storage are quite similar to the ones of previous services, nevertheless it's worth it to emphasise them.

Benefits

- Has no boilerplate, except a configuration file.
- Fast and secure to use
- Uploads and downloads are performed robustly regardless of network quality.
- Cloud Storage has great scalability.

Conclusion

In this research document we looked at Firebase, which is a platform developed by Google, as well as we dove into some of the Firebase services that I happened to use in my own project.

As the research clearly shows Firebase is definitely worth consideration when you decide which tools you are going to use while developing your application of any scale. Even for people that want to develop something for themselves but on behalf of a big company, Firebase offers a great range of services free of charge. After the research it becomes clear that using Firebase you will both decrease an annoying boilerplate code that would take you several days to write and at the same time you will get a significant level of security, speed and quality in your application.

Sources

Firebase:

1) Firebase Documentation (google.com)

2) What is Firebase? | A Quick Glance of What is Firebase with Application (educba.com)

3) Firebase - Wikipedia

4) Firebase Authentication - Javatpoint

JWT:

1) What Is JWT? How Does It Work? | Akana by Perforce

2) JSON Web Token Introduction - jwt.io

3) How JSON Web Token(JWT) authentication works? (sureshdsk.dev)