



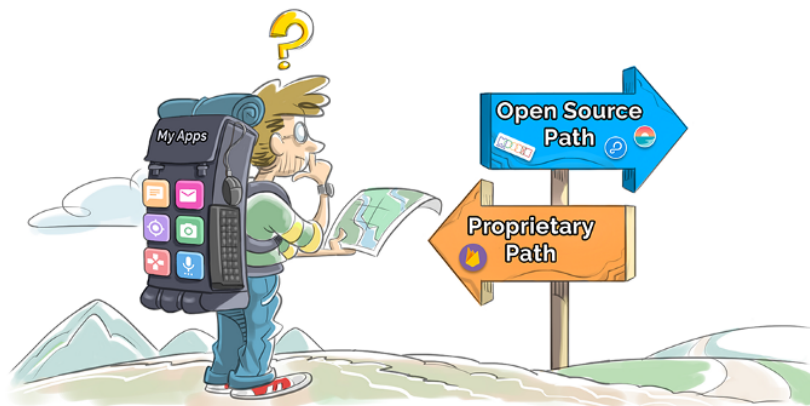
Brenda Clark

[Follow](#)

Tech addicted and Software Architect

Dec 1, 2017 · 8 min read

Firebase Alternative: 3 Open-Source ways to follow



Firebase Alternative: Open Source Parse

Firebase is a Google-backed, real-time backend service that empowers developers to build real-time mobile or web applications without complex protocols or rewriting the app from scratch.

It comes with many benefits such as fast and real-time updates, free tier up to 100 simultaneous connections, push-notifications support, easy setup, Robust APIs for JavaScript (including several frameworks like Angular), iOS & Android support, built-in auto-scaling, cross-platform API, and more.

Despite the fact that it's a great way to get your app up and running quickly (and easily of course), just like any other backend service, it does have some limitations which cannot be overlooked. For instance, unless your app runs on one centralized database and is updated by a vast quantity of users, it's a major overkill.

What are the downsides of Firebase?

One of its biggest and most obvious concerns is vendor lock-in, which prevents you from making your app portable. Besides that, you cannot

access your data as it is hosted on the Firebase server.

Talking about **vendor lock-in**, it is one of the greatest obstacles to enterprise cloud adoption. To be precise, it is a situation where customers are locked-in on single cloud provider technology (i.e. vendor) for products and services and cannot switch to a different vendor without incurring substantial costs, legal constraints, or technical incompatibilities. It involves a lack of compatibility between different systems, which compels a customer to continue using products from a particular vendor. It has several disadvantages, such as:

- You cannot optimize your backend according to your app's needs;
- You'll lose the ability to shop around and the power to negotiate;
- You'll not be aware of the new technologies that are used by other vendors;
- You are always locked by Google updates, and considering the nature of these updates, it is impossible for you to customize your app on server levels;

Another downside of Firebase is its database, which can be easily disorganized, and thus, form a tree-shaped structure, which can be hard to navigate and search. Also, there's a high potential for data loss. To be exact, you may often delete your entire database accidentally when you're in the console. Remember, two or three wrong clicks and all your data is gone! Also, the data validation does not support complex objects directly, so you'd need to validate individual child nodes separately.

Additionally, neither Firebase allows easy integrations with other apps nor is it possible to install your own modules on your backend. As far as scaling is concerned, even a fully managed database provider like Firebase is not a complete solution to all your scaling related concerns, as it becomes extensive once you scale your app. That's why it's always a better idea to choose an open source framework over Firebase for scaling apps.

So, if you're looking for a more complete backend solution (open source alternative) that not only provides similar functionality like Firebase but also gives you much more control over your data, we've

got you covered. In this article, we've disclosed three of the best alternatives to Firebase.

So, without wasting no time, let's dive in!

1-Parse Server



Overview: It's no secret that Parse.com it was the best Backend as a Service in 2016 when facebook decided to turn it off. Parse Server (the open source version of Parse) was created by facebook (and the open source community) with the same power and a huge and important differential: to be Open Source.

Parse Server comes with a gazillion out-of-the-box features that allows you to get your MVP out quick and effortlessly. Currently, Parse server is the most popular and robust BaaS framework available that helps developers build mobile apps faster without any technical locks. It is an open source version of the Parse backend that can be easily **downloaded for free** on GitHub. Many developers have successfully migrated their apps from other platforms to Parse Server.

Presently, the Parse Server is quite active, with 143 contributors and **more than 14k stars on GitHub**. The key attribute that sets Parse Server apart from Firebase and other fellow app development counterparts is that it can work with the Express Web Application framework, can be easily added to existing web applications, and can even run by itself, if required. Furthermore, it can be deployed to any infrastructure that can run Node.js.

Pros:

Allows you to easily customize the foundation of your app, to make it suitable for other OSs;

- Complete Dashboard to accelerate your backend development and manage your App easily;
- Provides you a very complete range of prebuilt SDKs to plug into your app project;
- Empowers you to make modifications to the API without having to write the code, and you can easily view analytics or push notifications as well;
- Parse Server is open source, which means you can have access to it anytime;
- Comes with a plethora of features that offer a better user experience;
- Can be hosted in almost any environment.

Cons:

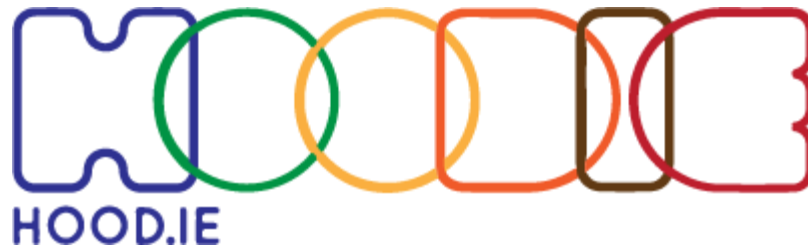
- Missing a stable command line interface to automate your custom code development;
- It's not multi-tenancy App compliant;
- No custom atomic operations.



Parse Server Made Simple

Back4App.com is the best implementation of Parse Server and solved these downsides. They have a very user-friendly CLI, it is possible to create as many App as you want and you can custom your Backend in many ways.

2-Hoodie



Overview: Hoodie is the First Offline JavaScript Backend that makes web application development very fast, easy, and accessible. It is a simple, self-hosted, and open source platform that allows developers to build robust and intuitive applications for both web and iOS. It utilizes a noBackend technology that makes the lives of developers unbelievably easy. Being a developer, all you need to do is deploy the Hoodie's backend to your server, and forget about it.

The key trait that gives Hoodie a competitive edge over Firebase and other app development platforms is the fact that it gives the control back in the hands of the frontend developer by abstracting away the backend completely. Yes, you heard it right! With Hoodie frontend development, without worrying about your server side, you just need to focus on its simple, easy-to-learn-and-implement frontend API. Unlike other backend platforms, Hoodie uses a community-driven approach to API designing, known as Dreamcode.

One of Hoodie's primary goals is to help users overcome their crippling dependence on the internet. With Hoodie, you can do it all—post comments, synchronize data, handle payments, edit accounts, send emails, and more without worrying about having a connection.

Pros:

- It has a noBackend, Dreamcode-driven API, which makes Hoodie code a breeze to write and read;
- NoBackend means you don't need a back-end developer;
- It's the biggest advantage is its offline support, which solves the major problem of connectivity dependence;

Cons:

- Going offline can be quite frustrating sometimes, as you can't have access to your Google Docs.
- It has a very small developer community and a very few of Firebase's trademark features.

3-Horizon



Overview: Horizon is a real-time, open-source backend platform that allows you to rapidly build and deploy web and mobile apps using a simple JavaScript API. Built by the RethinkDB team and an open-source community, Horizon lets you scale your app to millions of users without any backend code. It is a standalone server for browser-based JavaScript apps. Here's the best part: whenever you feel like you're ready to write custom backend code, you can easily load the Horizon modules directly in Node.js.

Horizon is made up of three key components:

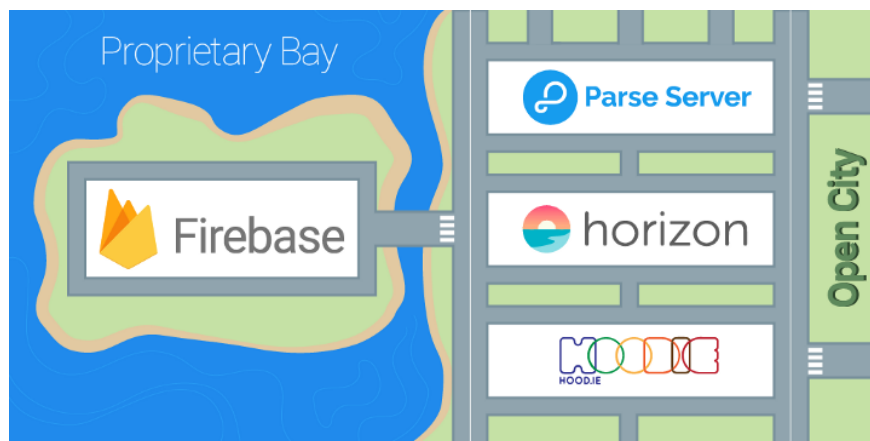
1. A middleware backend server built with Node.js and RethinkDB that supports data persistence, user authentication, input validation, real-time streams, and permissions;
 2. A JavaScript client library that frontend developers can use for storing JSON documents in the database, performing queries, and subscribing to live updates;
- A command-line (CLI) tool that can help you start up a local Horizon development server, generate project templates, and deploy your Horizon app to the cloud.

Pros:

- Built on RethinkDB, Horizon not only allows you to host your data but also do complex queries;
- You can build and deploy cross-platform, real-time web and mobile apps at lightning speed;
- Horizon is open source, which means you can run it on your laptop, deploy it to the cloud, or deploy it to any other infrastructure if required;
- Unlike Firebase, you can build complex enterprise apps with Horizon, not just basic ones with limited functionality;
- It reduces the amount of friction that developers face while building and scaling Web applications;
- It eliminates repetitive boilerplate and tedious steps such as hand-writing CRUD endpoints, authentication, and session management, and this is what makes it different from Firebase and other app development platforms.

Cons:

- It lacks numerous useful features that Firebase provides;
- Some of Horizon's flagship features are less mature than other platforms;
- Unlike Firebase, it has a small developer community;
- The Horizon team are refining some features and incorporating further improvements, because of which developers might encounter some rough edges while writing code;
- The long-term roadmap is still evolving;
- The Horizon code base is not as stable as Firebase or Hoodie.



Firebase Alternatives

Final Thoughts

These are just three of the many open-source alternatives to Firebase. Being a developer, you don't want your backend to be a deadlock for your app, which is why it is always advisable to decide wisely. One additional risk to consider while choosing an app development platform is longevity. Remember, if the service you're about to choose is not around for long, possibilities are that you'll have to find another alternative and migrate again. Also, bear in mind how easy it is to migrate data from that specific platform or stack. While considering the costs upfront is important, take a look at how those costs will ascend as your app scales.

Here's the most important thing to consider—always choose a technology that you're already comfortable with or looking forward to learning, forget the rest. After pondering all these options I decided to go with **Back4App**.

