HomeAutomation Follow
A Home Automation Startup.
May 16 · 10 min read

Firebase Costs Increased by 7,000%!

Update (05/17/2017 8:50 AM):

> *I am happy to say that I received a phone call promptly this morning from the executive team at Firebase. They professed their sincere apology for the situation and explained what happened specifically with the account.*

> *While we haven't reached a final conclusion on everything, it would appear the specifics of the article have been heard. Many of the complaints (namely the metrics, among other things) were outlined as things they would put a focus on to improve upon quickly.*

> *I will make sure to keep it updated as any relevant information is provided.*

Update (05/17/2017 1:00 PM):

> *Upon further discussions, credits have been taken care of and a plan has been set to address the concerns aggressively. Andrew (Founder of Firebase) and his team have been very proactive and has posted his comment on the situation in the bottom of the page.*

> *While it's never ideal to end up in a situation that you feel the need to express your situation publicly, the handling of the situation does reflect positively and has given us room to breath again.*

> *You can read Andrew's Comment Here.*

Summary

Due to a change in how they report data usage, our monthly costs for Firebase, a SaaS provided by Google, has increased from $25 a month to what is now moving towards the $2,000 mark—with no changes to our actual data use. This change was made without warning.

Don't get Trapped by the SaaS model—learn from our mistakes and be sure to implement workarounds so this doesn't happen to you!

> *Please let us know in the comments if you have any ideas, suggestions, or comments that could help us or any readers that may find themselves in a similar predicament.*

The Beginning

Like many, our startup started from something simple. It was just a tool to help *Home Automation Programmers* troubleshoot and integrate their devices and save them time.

Released at no charge, it quickly started to grow in popularity. It was exhilarating! We went from a Skype group of 10 beta testers to hundreds of active users, then thousands in only a few short months.

What a high it has been! I remember staring at our Google Analytics and Woopra page; following users as they navigated around.

Now I'll be the first to admit, **we made some major mistakes during this stage** (*Mistakes I hope we can help you to avoid)***.** We were building and adding new features and just trying to keep up with the requests that were flowing in.

Our mistake was not that we failed to read documentation. It wasn't that we picked services which had high failure rates or problems. It was one simple but absolutely critical mistake. One which I firmly believe a huge ratio of apps are making this very moment: **"Service Trap."**

We got ourselves into a situation that when a service we relied upon changed how they do things, our *bill was suddenly increased by over 7,000%.*

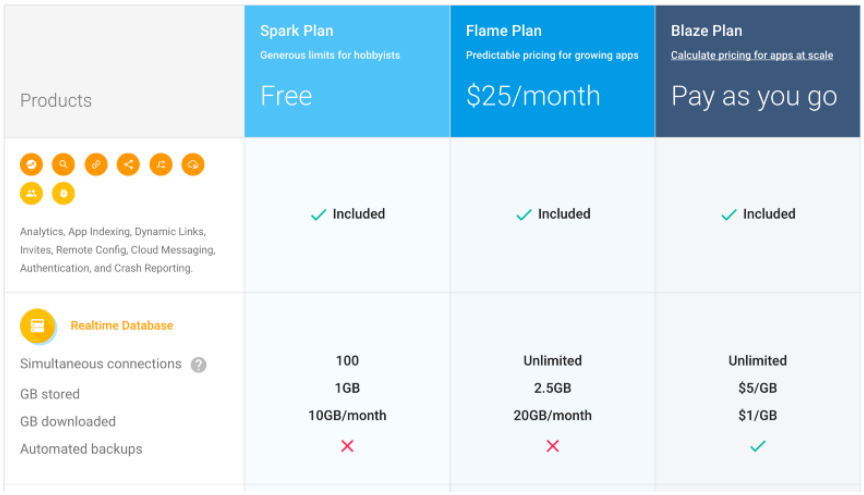
We simply have no solution which could be implemented in a short enough span to stop our costs from reaching tens of thousands of dollars. We are a self-funded bootstrapping startup, after-all.

Trapped! Our Luck has Run Out

It all started at the beginning of April (2017). We had been on the $25 monthly Firebase "Flame" plan since they moved to Google. We started using the Firebase service well before Google purchased them and turned it into the star of their mobile cloud initiatives.

It was only a few days into the billing cycle and *I got a notice that my app was going to be shut down due to excessive bandwidth. What!?* Alarmed, I logged in to see what might be happening.

Firebase's "Flame" plan provides 20GB monthly download bandwidth for $25 a month. It is a set plan so that overages don't surprise you which is nice. Well we were only a few days into the cycle and it stated we had already used 30 GB.



Products	Spark Plan Generous limits for hobbyists Free	Flame Plan Predictable pricing for growing apps $25/month	Blaze Plan Calculate pricing for apps at scale Pay as you go
Analytics, App Indexing, Dynamic Links, Invites, Remote Config, Cloud Messaging, Authentication, and Crash Reporting.	✓ Included	✓ Included	✓ Included
Realtime Database			
Simultaneous connections	100	Unlimited	Unlimited
GB stored	1GB	2.5GB	$5/GB
GB downloaded	10GB/month	20GB/month	$1/GB
Automated backups	✗	✗	✓

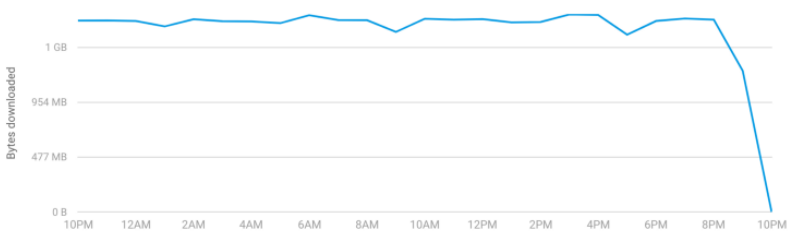
https://firebase.google.com/pricing/

Obviously our app being shut down and our users being cut off was not an option. Accepting the extra $10 overage for the 10 GB of data wasn't a huge deal. We promptly messaged support about the issue then switched to the "Pay as You Go" plan which is the only other option they have.

Upon pressing enter, our reported usage *had jumped to 100 GB*. You can imagine my concern. I shrugged it off and decided I would just figure it out with Firebase Support the next day.

When I got into the office the next morning, *we had suddenly jumped to 180 GB of download.* At $1 per GB you can imagine how scared I became as I calculated what this meant my bill was going to be. *What was happening!? Why!? How do I fix this?! What can I do?!* This was going to become thousands of dollars a month in unplanned expenses!

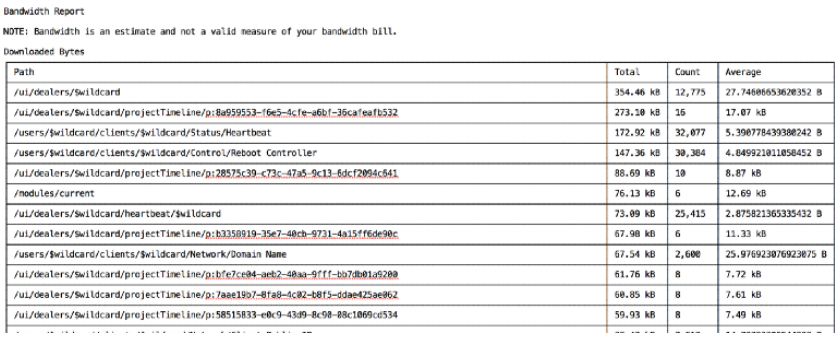
Firebase does not provide you with analytics or reasons for your usage — all you get is a blue line on a graph that says you have used this much data.



The only Analytics for Bandwidth you get with Firebase.

This didn't make any sense—how am I suddenly consuming so much bandwidth. Why? Where did it come from? Unfortunately there is really no way to know with Firebase.

Alas, I discovered they have a database profiling tool. So I ran the tool for a hour and it showed I used only a few MB of bandwidth—yet their analytics showed 2 GB for that same period!



Bandwidth Report

NOTE: Bandwidth is an estimate and not a valid measure of your bandwidth bill.

Downloaded Bytes

Path	Total	Count	Average
/ui/dealers/$wildcard	354.46 kB	12,775	27.748066536203352 B
/ui/dealers/$wildcard/projectTimeline/p:8a959553-f6e5-4cfe-a6bf-35cafeafb532	273.10 kB	16	17.07 kB
/users/$wildcard/clients/$wildcard/Status/Heartbeat	172.92 kB	32,077	5.390778439380242 B
/users/$wildcard/clients/$wildcard/Control/Reboot Controller	147.36 kB	30,384	4.849921011058452 B
/ui/dealers/$wildcard/projectTimeline/p:28575c39-c73c-47a5-9c13-6dcf2094c641	88.69 kB	10	8.87 kB
/modules/current	76.13 kB	6	12.69 kB
/ui/dealers/$wildcard/heartbeat/$wildcard	73.09 kB	25,415	2.875821365335432 B
/ui/dealers/$wildcard/projectTimeline/p:b3350919-35e7-40cb-9731-4a15ff6de90c	67.98 kB	6	11.33 kB
/users/$wildcard/clients/$wildcard/Network/Domain Name	67.54 kB	2,600	25.976923076923075 B
/ui/dealers/$wildcard/projectTimeline/p:bfe7ce04-aeb2-40aa-9fff-bb7db01a9200	61.76 kB	8	7.72 kB
/ui/dealers/$wildcard/projectTimeline/p:7aae19b7-8fa8-4c02-b8f5-ddae425ae062	60.85 kB	8	7.61 kB
/ui/dealers/$wildcard/projectTimeline/p:58515833-e0c9-43d9-8c98-88c1069cd534	59.93 kB	8	7.49 kB

A copy of the hour-long profile showing the downloaded bytes for the period. Nice warning at the top showing it is not a valid measure—— yet the difference is in the hundreds of thousands of percent!

I eventually got in touch with Firebase's Database team whom informed me that they changed how they report bandwidth to include the SSL overhead of requests. This includes failed attempts which are blocked by their security rules.

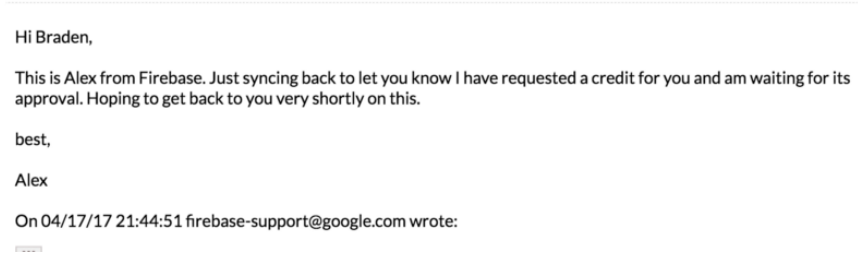
It was on this call that they informed me that their profiling tool does not show this SSL overhead. **In my case while every tool I have available shows absolutely no problem, my bill has increased by 7,000%!**

They said in most cases it doesn't make a big difference… unless you use the REST API. In our case, our applications original code was in an unsupported language so we used the REST API to compensate. All it did was read a boolean value every minute to check if it needs to process anything further.

Something that for 2 years was not an issue, yet suddenly our bill went through the roof.

Upon further investigation, they told me that the excessive usage was due to requests not using something called *TLS Tickets* (something I have never seen mentioned in any library or package) (**Update**: we are using TLS Tickets/Resumption with the requests) and not setting "Keep Alive" to true.

There is no documentation about this anywhere**.** Likely because this was not something that would cause an issue until they changed their billing. We read the documentation, wrote the requests based upon it. I have personally read all of their documentation at least 30 times.


Hi Braden,

This is Alex from Firebase. Just syncing back to let you know I have requested a credit for you and am waiting for its approval. Hoping to get back to you very shortly on this.

best,

Alex

On 04/17/17 21:44:51 firebase-support@google.com wrote:

The last response Firebase has sent me—over a month ago. No replies to my messages since, thousands of dollars charged.

At first they seemed to be willing to work with us. They set up a time to discuss and assured us that we would be taken care of as we work to resolve this problem.

We got on the phone to discuss everything and they said they would take care of it, giving us time to fix things. They emailed and let us know they were on top of it and would get credits sorted out.

- **That is until my card was charged.**
- Until they stopped responding to our emails.
- Until they completely ignored us and gave us absolutely no recourse other than to dispute the charges—*which would of course shut down the service completely and kill our app and scare away our users.*
- *They have no phone number to contact,* no way to dispute this other than email—which they have ignored us for over a month now without replying to our continued requests. *Trapped. Doomed. We have no further options.*

So in summary:

1. They changed how they report their bandwidth usage, increasing our bill by 7,000% without any change to our actual usage. *After years of using the service.*
2. No warning or message was sent out that this was being done—we only got notified once they were planning to shut down our app completely.

3. Their profiling tools do not show the increased usage. You can only see it by looking at your massively increased bill.

No Path but Down

Ok so it seems like a simple enough thing to fix. You have one little thing that is doing something, so change it! Well if you said that to us today, with our current architecture, I would say “ok, give me a hour”.

We never hard code URLS now. Instead we rely on proxies which we can modify as-needed. We implement API’s in-between 3rd parties that allow us to switch to a similar service if needed, and are learning better tactics daily! We are a small startup and self-funded—so **would welcome your ideas, comments, and suggestions!**

Unfortunately, it’s not that simple here. You see, this was one of the first lines of code that we ever wrote while initially integrating with the Firebase Service. It’s hard-coded into our “version 1” app (***our biggest mistake***) and deployed to thousands of systems across the globe. It was written at a time that the app was just a fun idea. It had never been an issue for years, and to this day all of the profiling tools available do not report the issue. Our bill has been at a steady $25 or lower since inception.

However, since this is in the first version of our app (which is downloaded locally to Home Automation Systems in almost every country of the world), we can’t do anything short of shutting down the entire service while we implement our new version… something that would surely kill our user base immediately as the effects would be catastrophic causing many to travel hundreds of miles to their clients homes across their cities to resolve.

As we grew we quickly realized that being trapped into using any specific service might cause an issue like this. However, I never in my wildest dreams thought it would be this significant and this sudden.

We have since changed our model to one that allows us to interchange things dynamically via a serverless architecture. One that fixes all of our issues, **but one that we can’t fully implement at this point in time to cost us any less than $5,000–10,000 in unexpected Firebase service costs that we unfortunately can not afford.**

So please, I beg of you… learn from our embarrassing and now dooming mistake:

- Always build your architecture in a way that will avoid becoming trapped into a specific service. Build your application in a way that swapping one service for another is as simple as possible. **Edit:** As users have pointed out in the comments, a few ways to do this would be: Alex R. Young suggests: *"Wouldn't relying on AWS Lambda potentially cause the same problem? Why rely on any specific vendored service when you can run services in containers or VMs that can easily be redeployed to any of the big 3's architectures?"*

- Always keep in mind that the services you use can change at any moment and put you in a situation where you don't have options if you aren't careful.

- Whenever possible, rely on your own infrastructure. The SaaS model seems attractive to both Startups and Service Providers…. but in the end, its the Startup that gets bitten by it and the providers that make the real money.

- Always heavily consider open source alternatives (something which didn't exist for Firebase at the time… but now alternatives like Horizon and Backendless exist, for example.

Conclusion

I can not believe I am writing this article. I am not the kind of person that would write a public complaint or story such as this. Like many of you, we spend long hours learning, planning, coding, and implementing our dream startup. We ride the rollercoaster; the crazy ups and downs, the excitement, and the stress that comes along with it.

I am honestly extremely embarrassed that this has happened to us. It is easy to be overwhelmed by the immense knowledge many seem to have in the digital world. It was extremely hard to write this article for fear of being looked down upon for our mistakes. I can only hope that some of you may learn from our embarrassing mistakes and implement solutions to the growing problem of service trap.

Taking the leap, quitting your job, and deciding to make this crazy little idea you have into a career is easily one of the scariest and most exciting things you can do. Without hesitation I would compare that moment to any other experience I have had in my life.

Of course as we are learning, we all make mistakes. Unfortunately, it can be difficult to see that some of these *"minor"* mistakes can come back to truly bite you, **especially when a service decides to change how they do things without any warning.**

Not Here to Put Them Down

I don't want to speak badly about any service or platform.

They didn't do this out of spite, or to hurt us. It is the result of a ***"small change"*** on their end becoming a ***"massive change" (by a factor of over 70x)*** to our monthly bill. One that saw us going from $25 to a whopping $1,750 or more (its still growing).

I do, however, want to try to **help you from making the mistakes we did**. Mistakes which are now going to cost us our future as changes to how Firebase reports their bandwidth has increased our bill by 7,000%.

Do yourself a favor. Take any necessary time to consider the services you are signing up for and implementing. Consider how those decisions might change your fortunes in the future from a momentum-building excitement to a complete crash overnight.