



Sarah Mei @sarahmei Mon Jan 11 03:06:13 +0000 2021

Perhaps we're all tired of dunking on parler, but just in case you're not, I did some investigation this evening.

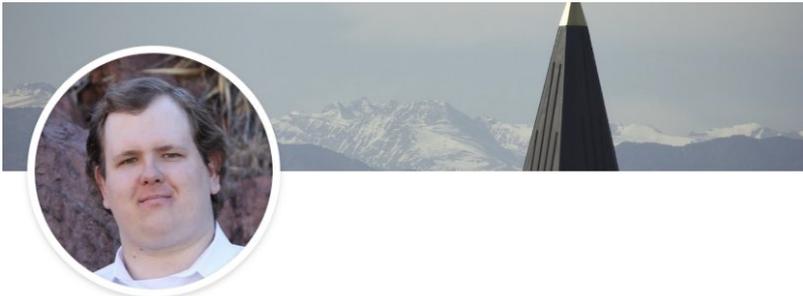
tl;dr: technical clown shoes.

■

Their CTO is a person named Alexander Blair, whose career (according to his LinkedIn) has been mostly sys admin and DevOps jobs. His job history goes back to 2008. <https://t.co/GAPBMmJyIg>

On the surface, Blair is not an unreasonable CTO choice for a small startup thinking they're going to have backend scaling issues.

(Interestingly, there's another person on LinkedIn with CTO of parler in his headline - cofounder Jared Thomson. On parler itself Thomson calls himself "cofounder & lead developer", so perhaps Blair was brought in to be the adult once parler got going. Speculation on my part.) <https://t.co/qpT9cxVXU5>



Jared T. · 3rd

Founder, CTO, and Backend Software Engineer
at Parler

Parler · University of Denver

Henderson, Nevada, United States · 50 connections

ANYWAY! On parler itself, Blair gives us a little more info about what he's into. He identifies himself with that most serious of technical groups - the blockchain enthusiasts. ■ <https://t.co/8NDdVD0XIn>



Alexander Blair
ablair

Software and Blockchain Engineer,
Emergency Technical Fire
Extinguisher, and Infrastructure
Architect. Chief Technology Officer
@Parler

But whatever - everyone has their flaws. And in all seriousness, parler needs an infrastructure architect. They need people who know about things like, oh...how databases work. And stuff.

Cut to five months ago. Parler has a significant outage - significant enough that it's called out individually in this article about how parler's performance is struggling. <https://t.co/fK0MFndz82>

This is due to their own success. A few months prior, Twitter had started labeling some of Trump's tweets, and parler started pushing their service as an alternative. They used the unfortunate hashtag [#twexit](#).

Blair comes on to parler to explain the outage. Here's where the clown shoes start being apparent. I'll let him tell it in his own words... <https://t.co/J0NeytDZEf>



Alexander Blair · @ablair

5 months ago · 254

Good Evening Everyone!

It's #parler #extra #latenight checking in again with things going on tonight here in UTC-7.

I'd like to extend our apologies to those who had issues with the site a few hours ago, I was, admittedly, dead asleep, and we'd missed how many notifications had been written to the database over time. (Over 2.1 billion, as the case may be. Yeeeeesh!). We've performed an audit of all tables that might run this risk (Votes, Followers, etc.) and patched them as needed

A couple of comments on feature requests, our Bugzilla instance is up, and we welcome feature requests, as well as bugs there, but I'll comment on a few here.

Echoing Comments - This is something we could like to do in general, however, this requires some serious thought and discussion on how to preserve the context.

Editing - This is being discussed on the Bugzilla some, and I'll be adding some comments and thoughts there.

Bugzilla link in comments!

Hope everyone has a good day/night/etc!

69 35 106

Hide Comments

Perhaps this isn't quite obvious enough. Here's a followup comment he made: <https://t.co/J3ugC9N1ul>



Alexander Blair · @ablair

5 months ago

Every notification is stored with a unique ID in the database, over time, we've written over 2.1 billion notifications to the database, which caused a limit we weren't actively aware of to get hit, blocking the writing of new notifications, and causing the weirdness on the site.



Does that number 2.1 billion seem familiar at all?

If you've worked with enough relational data, you'll recognize it as the limit of an integer auto-increment primary key. In other words, if you make a table in a database with an integer primary key, you can only put 2.1 billion records in it.

I remember when this happened to the tweets table at Twitter. It was identified and planned for months ahead of time. So just to be clear here:

- 1) their primary data store is relational
- 2) they put integer PKs on everything
- 3) they didn't realize that the PKs limited the size of the tables
- 4) when it fell over, only one person could fix it - Blair - and he was asleep.

ALL FOUR of these are total clown shoes for a social network - even a small one.

And especially for a CTO who's an infrastructure specialist. I mean, COME ON.

A social network that depends on a relational store is just...bananapants. Showing a feed is like a nine table join - people x posts x permissions x avatars x comments x likes x shares x (etc).

So there's that, but if we're being charitable, perhaps Blair inherited that decision from Thomson.

It still doesn't excuse Blair's total lack of awareness of how primary keys work. He called it "a limit we weren't actively aware of."

And then there's the problem that ONLY Blair could fix this. Perhaps they've hired more backend folks since then, because if not, they have a -30% chance of ever reappearing on the internet.

I just find this whole thing ridiculous. And hilarious.

Don't get me wrong - there was a point even several years into my career where I did not understand this nuance.

Fortunately, at no time when I was so ignorant was I a CTO

They might have done better with four ferrets in a trench coat. [fin]