

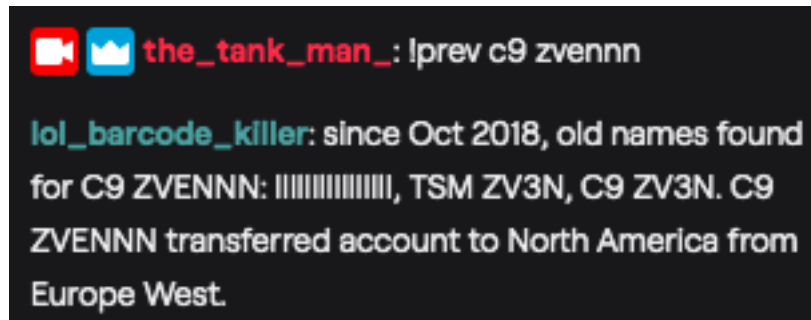


## 4 Implementation

I originally implemented the bot on glitch.io using JavaScript and Node.js, and it was great except it constantly stopped working. For one thing, dev API keys expire every 24 hrs, so I had to get a permanent personal key from Riot for the project (which is still rate limited but doesn't need to be refreshed).

That made it easier, but glitch still kept putting it to sleep. So I moved it to AWS, which made it slightly more annoying to get npm and everything again, and still seemed to stop working when I quit SSH, but once I started it in a tmux terminal then disconnected then quit it kept working.

It turned out that there were issues with accounts that had transferred from different regions, so in addition to recording old names in NA games I had to fix that:



I also had to make it so that it wouldn't start a new lookup while one is in progress to prevent multiple people from trying to use it at once and making calls to the API too quickly.

The code is still available on [glitch](#) if you want to play with it.

## 5 Future Work

What really costs us is calls to the API, so it might be possible to get a theoretically better complexity by using a binary rather than linear search to look for different names, but I'm not sure this would actually help, not just because of a constant factor but because the match history API only supports extracting 100 consecutive games at a time. It might help in some cases with really long match histories though, with the caveat that if a player started with name A, changed to name B, then went back to name A, it wouldn't catch it. But probing every 50 or whatever games might miss brief changes too.

Another way of potentially making lookups faster would be caching names by account id to save API calls if the same player is looked up multiple times.