# Rage-quit: Coder unpublished 17 lines of JavaScript and “broke the Internet”

*Sean Gallagher - 3/24/2016, 10:10 PM*

**Biz & IT —**

## Dispute over module name in npm registry became giant headache for developers.

Photo illustration by Aurich Lawson

It all started with a request from the developers of a messaging application to an open source developer to change the name of a library. It ended with JavaScript developers around the world crying out in frustration as hundreds of projects suddenly stopped working—their code failing because of broken dependencies on modules that a developer removed from the repository over a policy dispute.

At the center of it all is npm, Inc., the Oakland startup behind the largest registry and repository of JavaScript tools and modules. Isaac Schlueter, npm's creator, said that the way the whole thing shook out was a testament to how well open source works—another developer replaced the missing link quickly. But many developers are less than elated by the fact that code they've become dependent on can be pulled out from under them without any notice.

The disruption caused by the wholesale unpublishing of code modules by their author Azer Koçulu

was repaired in two hours, Schlueter told Ars, as other developers filled in the holes in the repository. The incident is, however, prompting Schlueter and the team at npm Inc. to take a look at how to prevent one developer from causing so much collateral damage.

To understand how one developer's rage-quit from a JavaScript code registry could suddenly cause all sorts of things across the Internet to begin to fail, you need to understand the strange nature of npm, which is inextricably tied to node.js—a popular open source tool that allows developers to write Internet server applications (as well as desktop and other types of applications) in JavaScript. Node.js uses npm as its default "package manager" for installing software, much as Linux distributions use apt-get.

While the tools for npm are open source, the global public registry that it taps into is the service of a private company with venture capital backing. And npm, Inc., which aims to make revenue off private registries, treats the global public registry as an editorial product—a product that many developers have become dependent on to tap into a vast, automatically updated collection of open source code.

It's an arrangement that has worked well, largely—except when it doesn't. And this week, it suddenly didn't.

## Kik in the ass

The roots of the problem were in a project Koçulu, a prolific developer of open source JavaScript libraries and a longtime unofficial evangelist for npm, had launched on npm called "kik"—a command-line tool and library for "kick-starting" the setup of development projects, including their Git remote repository. Koçulu registered modules for the project on npm. But in the eyes of developers at Kik Interactive, the developers of the Kik mobile messaging app, this was the moral and legal equivalent of domain squatting—it prevented them from registering modules for developers to use with their trademarked name.

In an e-mail to Koçulu on March 11, Bob Stratton [**Update:** Stratton is a contracted patent agent for Kik] explained the issue. "We're reaching out to you as we'd very much like to use our name "kik" for an important package that we are going to release soon," Stratton wrote. "Unfortunately, your use of kik (and kik-starter) mean that we can't and our users will be confused and/or unable to find our package. Can we get you to rename your kik package?"

Koçulu replied an hour later, simply saying: "Sorry, I'm building an open source project with that name."

This didn't sit well with Kik (the company). Stratton responded the next day, saying "We don't mean to be a dick about it, but it's a registered trademark." He then mentioned that if Koçulu went ahead with a project with that name, "our trademark lawyers are going to be banging on your door and taking down your accounts and stuff like that—and we'd have to do all that because you have to enforce trademarks or you lose them. Can we not come to some sort of a compromise to get you to change the name without involving lawyers?"

"Hahah, you're actually being a dick," Koçulu replied. "So, fuck you. Don't e-mail me back." After a final plea from Stratton, he answered, "Yeah, you can buy it for $30,000 for the hassle of giving up with my pet project for bunch of corporate dicks."

At this point, Stratton and Kik pleaded with npm's support team to help straighten things out. He sent several e-mails to npm, asking the support team to intervene. Schleuter made the call to give Kik the name and tried to diplomatically break it to Koçulu, expressing sympathy with his frustration.

"We have a very well documented policy for handling these disputes," Schleuter told Ars. "It very quickly became obvious that they were not going to be able to resolve their dispute over the name. We made the decision based on what we thought would be in the best interest of the NPM community. What it came down to is that a reasonably well-informed user who types 'npm install kik' would expect to get something related to Kik. So that's why we turned (the name) over."

Koçulu did not take the decision well. "I know you for years and would never imagine you siding with corporate patent lawyers threatening open source contributors," he wrote back. Disillusioned, Koçulu demanded, "I want all my modules to be deleted including my account, along with this package. I don't wanna be a part of NPM anymore. If you don't do it, let me know how do it quickly. I think I have the right of deleting all my stuff from NPM."

Koçulu told Ars that Schleuter sent him a command to do just that. "The second email I got from NPM was the founder Isaac giving me a one-liner command that deletes all my stuff," he said in an e-mail. And he used that command, deleting 273 modules he had registered in npm (though he left the modules available through GitHub).

In a post on Medium, Koçulu said, "This situation made me realize that NPM is someone's private land where corporate is more powerful than the people, and I do open source because, **Power To The People**."

And that is when the JavaScript hit the dependency fan.

# Justify yourself

```
npm ERR! npm  v2.14.7
npm ERR! code E404
npm ERR! 404 Registry returned 404 for GET on https://registry.npmjs.org/left-pad
npm ERR! 404
npm ERR! 404 'left-pad' is not in the npm registry.
npm ERR! 404 You should bug the author to publish it (or use the name yourself!)
npm ERR! 404 It was specified as a dependency of 'line-numbers'
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.
npm ERR! Please include the following file with any support request:
npm ERR!     /home/travis/build/coldrye-es/pingo/npm-debug.log
make: *** [deps] Error 1
```

What thousands of JavaScript developers saw on March 22 in their logs.

One of Koçulu's deleted modules happened to be left-pad, a very widely used chunk of 17 lines of JavaScript code used to right-justify text. In fact, many people weren't aware that they were using it in their code, because it was buried in the dependencies of tools they used.

"Indirectly, there were a couple of very large packages that depended upon left-pad," Schleuter said.

"So when that disappeared from the registry, their builds started breaking and people got very upset."

One of those very large packages was Babel, the JavaScript "compiler"—a tool that cleans up and updates JavaScript code to match current standards. Babel uses another module, called line-numbers, that depended on left-pad. Suddenly, thousands of developers saw their code failing. And at this point, lots of people started freaking out. One developer declared, "This kind of just broke the internet."

Within ten minutes, as Schleuter describes in a blog post about the episode, developer Cameron Westland had stepped in and published a functionally equivalent version of left-pad. But it took a bit longer for all the collapsed stack of dependencies to be sorted out, since some of the code breaking specifically called a different version number than the one Westland had put on his left-pad. And the anger over the outage didn't end when everything was declared fixed.

Schleuter said that the speed with which the holes created by Koçulu's unpublishing were filled demonstrated the power of the open source community around JavaScript. "The open source community really was working—the system worked," he said. "Extremely quickly, the community came together and fixed the issue." But he acknowledged that many people were still upset that it had been allowed to happen in the first place—that someone had been allowed to arbitrarily yank code out of the system and break theirs. "'That's one of the things that's adding fuel to this fire," Scheluter acknowledged. "'Why do you let this happen? Why can people unpublish things and break my builds?' That's what a lot of people are really upset about."

And yes, it is. A discussion over a user request to kill npm's unpublish feature became heated, and when npm's command-line interface team lead Forrest L Norvell locked the discussion "because I want to have an evening away from this," it further fanned flames. The discussion thread has not yet been unlocked. James Nadeau wrote a long separate comment on npm's GitHub portal, entitled "Should I trust npm?", in which he expressed concerns many had raised:

> (T)here has been a series of decisions, commitments, and actions that this project's maintainers have taken that have eroded the trust of it's users.[sic]
>
> I can't trust that a package will always be available.
> I can't trust npm will keep a published package around.
> I can't trust they will respect my actions of unpublishing something from npm.
> I can't trust that project maintainers will at least listen to my concerns.
> I can't trust.....
> I imagine the number of people taking a look at how much they trust, need, and depend on npm right now is huge. I'm actively taking steps to mitigate how much I depend on this project to be available, and at what point in my development process I make use of it. I've talked to others doing the same.
>
> I'm taking actions that demonstrate my loss of trust with this project. In doing so, I can see multiple ways in which the npm organization is much less involved with the work I produce. This series of thoughts doesn't make me want to open up my wallet for you anytime soon. Quite the opposite.
>
> Is this what you want your community members to be thinking and doing right now?

Schleuter said that npm Inc. is "definitely taking a close look at how things work when you

unpublish a package, and what we need to change there to facilitate the smooth operation of the JavaScript community. That's our number one focus." He said there were "some historical reasons for letting people unpublish" and that he didn't have any details yet on what changes would be made—nor would he speculate about what those changes would look like.

This isn't the first time a developer has gotten angry with npm Inc.'s handling of the registry. And it certainly won't be the last. But this particular episode has underscored the risks associated with the stack of dependencies many JavaScript developers' code now sits atop.

**Update:** In an e-mail to Ars late on March 25, Koçulu assessed what had happened as the result of his removal of his projects from npm. "Feeling very sorry for interrupting people's work," he said. "I did it for the benefit of the community in long term. NPM's monopoly won't be dictated to the free software community anymore."