

Forum:[Hack Forums](#)

Thread: [Custom node modules / npm packages](#)

Hackers - October 25, 2023, 12:32 am

I created an NPM package earlier today to connect to one of my existing APIs. I added keys so I can let others use it and setup some funcs that call my API endpoints / edge functions.

While planning though, I realized how many reputable companies/services with APIs don't actually have their own NPM distributions. So naturally now I can't get off the idea of making some API wrappers for those & picking up views/stars/forks on Github or whatever else comes from being first to market with specific integrations.

I use JS for my packages but am planning on trying out a C++ module soon enough

Do any of you have experience with this? The whole process is super simple to be honest, and if you need help getting started just let me know :)

perm - October 26, 2023, 1:49 pm

How do you know that those companies don't use a custom setup?

Disease - October 26, 2023, 1:57 pm

(Oct 26, 2023 01:49 PM)perm Wrote: [\[CITE\]](#) How do you know that those companies don't use a custom setup?

I don't think he's referring to what they're doing in their own codebases, but rather what's published and available for their tools/APIs for use by the general public.

Hackers - October 26, 2023, 9:07 pm

(Oct 26, 2023 01:49 PM)perm Wrote: [\[CITE\]](#) How do you know that those companies don't use a custom setup?

Certain companies just don't have package releases for devs on various platforms, and there aren't always actual robust flexible libraries. Some enterprise services with web APIs but nothing you can just install with a package come to mind.

(Oct 26, 2023 01:57 PM)Disease Wrote: [\[CITE\]](#) I don't think he's referring to what they're doing in their own codebases, but rather what's published and available for their tools/APIs for use by the general public. Yep, exactly - I'm interested in essentially being first to market or first to a solid open source release, powered by certain APIs - some big, some small, it's amazing how many companies are running on old tech or otherwise not significantly improving their core software.

perm - October 27, 2023, 2:43 am

(Oct 26, 2023 01:57 PM)Disease Wrote: [\[CITE\]](#) I don't think he's referring to what they're doing in their own codebases, but rather what's published and available for their tools/APIs for use by the general public.

Most companies don't reveal their setup anyway, so it's just a strange conclusion he's making.

Disease - October 27, 2023, 6:26 am

(Oct 27, 2023 02:43 AM)perm Wrote: [\[CITE\]](#) Most companies don't reveal their setup anyway, so it's just a strange conclusion he's making.

That's not the conclusion he's making. You're just not following correctly.

Company A releases a new database for public use. But they didn't release an npm package to interface with it. The OP is talking about identifying those scenarios, and writing his own packages to do that for people to use. That way other developers will have an easier time at using said software by not having to write raw connections to it.

Hackers - October 27, 2023, 8:11 am

(Oct 27, 2023 02:43 AM)perm Wrote: [\[CITE\]](#) Most companies don't reveal their setup anyway, so it's just a strange conclusion he's making.

I'm not talking at all about internal setups or tools. I understand completely what you're saying, it's just I work with a number of enterprise clients and some are publicly traded. We have account reps at a lot of big service providers who are able to tell us exactly what's available, even non-public tools. We still don't get access to their internal tools, but for this I'm specifically talking about how many big companies do not publish public packages on NPM or other package managers even though there is nothing preventing them from being made.

I think that it is a mistake for certain companies to ignore this because ease of development can be a huge reason that people recommend a partnership to a big client. When a package is released, stable, and well-documented, it makes it much easier to produce accurate estimates, plan 1+ year scopes, etc. I think that there is a lot of profit, or open source notoriety that can be gained from releasing packages for certain companies, so that's my goal with that.

(Oct 27, 2023 06:26 AM)Disease Wrote: [\[CITE\]](#) That's not the conclusion he's making. You're just not following correctly.

Company A releases a new database for public use. But they didn't release an npm package to interface with it. The OP is talking about identifying those scenarios, and writing his own packages to do that for people to use. That way other developers will have an easier time at using said software by not having to write raw connections to it. Yep, 100%. 'npm install hf' versus installing Axios/fetch/etc and having to manually write out all the API connections, exceptions, tests, etc. Even if a repo codebase can be shared with manual connections to an API, having a stable package release that handles that is a great way to avoid spaghetti code, dev mistakes, and having to rewrite duplicate code across multiple projects.

perm - October 27, 2023, 8:41 am

(Oct 27, 2023 08:11 AM)nodes Wrote: [\[CITE\]](#) I'm not talking at all about internal setups or tools. I understand completely what you're saying, it's just I work with a number of enterprise clients and some are publicly traded. We have account reps at a lot of big service providers who are able to tell us exactly what's available, even non-public tools. We still don't get access to their internal tools, but for this I'm specifically talking about how many big companies do not publish public packages on NPM or other package managers even though there is nothing preventing them from being made.

I think that it is a mistake for certain companies to ignore this because ease of development can be a huge reason that people recommend a partnership to a big client. When a package is released, stable, and well-documented, it makes it much easier to produce accurate estimates, plan 1+ year scopes, etc. I think that there is a lot of profit, or open source notoriety that can be gained from releasing packages for certain companies, so that's my goal with that.

Yep, 100%. 'npm install hf' versus installing Axios/fetch/etc and having to manually write out all the API connections, exceptions, tests, etc. Even if a repo codebase can be shared with manual connections to an API, having a stable package release that handles that is a great way to avoid

spaghetti code, dev mistakes, and having to rewrite duplicate code across multiple projects.

Do you realize how often the big companies update things such as Wordpress plugins with millions of users in case a bug makes the entire plugin unusable and the fix would be very easy? Usually 8-12 months for the most of them. Having people rely on packages from big companies instead of open source is stupid, the companies won't dedicate the resources to maintain them especially in case of 0-days and it's also a legal risk because they can and will get sued when things break.

Disease - October 27, 2023, 9:38 am

(Oct 27, 2023 08:41 AM)perm Wrote: [\[CITE\]](#) Do you realize how often the big companies update things such as Wordpress plugins with millions of users in case a bug makes the entire plugin unusable and the fix would be very easy? Usually 8-12 months for the most of them. Having people rely on packages from big companies instead of open source is stupid, the companies won't dedicate the resources to maintain them especially in case of 0-days and it's also a legal risk because they can and will get sued when things break.

Right... which is why [Hackers](#) is talking about providing said packages. I'm not sure if you're actually intending to argue here, though your post is absolutely coming across that way. But hopefully you can recognize that you're actually agreeing with [Hackers](#) here....

perm - October 27, 2023, 9:47 am

(Oct 27, 2023 09:38 AM)Disease Wrote: [\[CITE\]](#) Right... which is why [Hackers](#) is talking about providing said packages. I'm not sure if you're actually intending to argue here, though your post is absolutely coming across that way. But hopefully you can recognize that you're actually agreeing with [Hackers](#) here....

Not really. Because having those companies provide said packages is a legal risk. And they would need to spend 95% of the dedicated resources on managing legal risk and 5% on actual development.

This is one of the reasons why some of the big companies provide financial and other support to open-source packages.

It's a similar reason why Wikipedia doesn't hire editors. Section 230 protections shield them from getting sued.

Hackers - October 27, 2023, 1:03 pm

(Oct 27, 2023 09:47 AM)perm Wrote: [\[CITE\]](#) Not really. Because having those companies provide said packages is a legal risk. And they would need to spend 95% of the dedicated resources on managing legal risk and 5% on actual development.

This is one of the reasons why some of the big companies provide financial and other support to open-source packages.

It's a similar reason why Wikipedia doesn't hire editors. Section 230 protections shield them from getting sued.

I know this, but there are also many who do have their own API packages. With graphql, standard rest protocols, it's actually very common.

I'm not claiming I'll make anything more than a cool Git repo here, and of course you are right that a large company may spend months or years maintaining a large library.

However, I'm building microservices as relatively small node modules that simply make it easier to connect with services that already exist, work, have auth security, error handling, Caching, and documentation.

I am not trying to do anything else but make some fairly simple connections to established backends in public repositories, etc. For fun, and if the public wants to fork or build off my core, that's great. If the company or a client of that company notices it and does 1 deal w/ our team, it pays off for me. I'm just building tools for IOT, web, etc, and nothing more or less for these types of side projects.

Disease - October 27, 2023, 1:55 pm

(Oct 27, 2023 09:47 AM)perm Wrote: [\[CITE\]](#) Not really. Because having those companies provide said packages is a legal risk. And they would need to spend 95% of the dedicated resources on managing legal risk and 5% on actual development.

This is one of the reasons why some of the big companies provide financial and other support to open-source packages.

It's a similar reason why Wikipedia doesn't hire editors. Section 230 protections shield them from getting sued.

Companies don't necessarily opt out of publishing and maintaining these kinds of packages / tooling out of legal repercussions. That might be a concern for some. But it's not the prevailing cause. Look at the likes of MySQL, for example, who writes and publishes on Github connectors for virtually every major language. Then the community takes that and writes up more abstracted interfaces to solve particular use cases.

If they provide such resources under copyleft agreements, which they're generally always going to do as it'll be open source, they're free of guarantee and warranty. No copyleft license provides provisions where it would be permissible for an end-user to sue over problems in using the solution. So that's just not an issue 99% of companies actually have to consider.

It's more to just do with their perception of necessity and value.

The reason big companies (and small ones too) pledge money for open source projects is because of that value proposition alone. They might use a package in their ecosystem and recognize that it remaining in existence is important for them. They may have the capability of taking over maintenance of that software if the current maintainer(s) stopped. But until that happens, it may be seen as being more cost-effective to provide some monetary pledge rather than divert limited engineering hours to it.

It's just economics. It's not a legal consideration.

perm - October 27, 2023, 2:00 pm

(Oct 27, 2023 01:55 PM)Disease Wrote: [\[CITE\]](#) Companies don't necessarily opt out of publishing and maintaining these kinds of packages / tooling out of legal repercussions. That might be a concern for some. But it's not the prevailing cause. Look at the likes of MySQL, for example, who writes and publishes on Github connectors for virtually every major language. Then the community takes that and writes up more abstracted interfaces to solve particular use cases.

If they provide such resources under copyleft agreements, which they're generally always going to do as it'll be open source, they're free of guarantee and warranty. No copyleft license provides provisions where it would be permissible for an end-user to sue over problems in using the solution. So that's just not an issue 99% of companies actually have to consider.

It's more to just do with their perception of necessity and value.

The reason big companies (and small ones too) pledge money for open source projects is because of that value proposition alone. They might use a package in their ecosystem and recognize that it remaining in existence is important for them. They may have the capability of taking over maintenance of that software if the current maintainer(s) stopped. But until that happens, it may be seen as being more cost-effective to provide some monetary pledge rather than divert limited engineering hours to it.

It's just economics. It's not a legal consideration.

I agree but disagree with your conclusion that it's not a legal consideration. Look at this link how crazy people interact with Wikipedia and why they don't have editors; they are protected by Section 230. But noobs will still threaten to sue every time someone writes something they don't like.

<https://www.eff.org/issues/cda230/successes/wikipedia>

If Google or Meta for example, releases lots of packages that have severe errors, and some people get hacked because of poor firewall settings, you bet some noobs will come at them with class-action lawsuits.

Hackers - October 27, 2023, 2:06 pm

(Oct 27, 2023 02:00 PM)perm Wrote: [\[CITE\]](#) I agree but disagree with your conclusion that it's not a legal consideration. Look at this link how crazy people interact with Wikipedia and why they don't have editors; they are protected by Section 230. But noobs will still threaten to sue every time someone writes something they don't like. <https://www.eff.org/issues/cda230/successes/wikipedia>

If Google or Meta for example, releases lots of packages that have severe errors, and some people get hacked because of poor firewall settings, you bet some noobs will come at them with class-action lawsuits.

That's true, which may also be why there are so many projects involving custom Google integrations. There's lots of frontend packages for maps, backend wrappers of Google services, etc.

I think the guarantees of enterprise SLAs for core backend APIs are where the risks are taken, so I can understand your point that many companies do not want more risk of managing an external, oftentimes platform specific library. Which is the use case for me, or you, or others to build out useful packages that have clear scopes.

That said, plenty of companies do release packages that they are not legally responsible for because it may just be a development example with a disclaimer that it they don't guaranteed its functionality in production environments. However, plenty do produce good codebases for plenty of different stacks. Sometimes they won't maintain them, but it can significantly help custom development along the way.

--

ITT / TLDR:

I think it's a benefit to developers that certain niche companies don't have public packages available. This is less of a dig at them and more about the opportunity to build cool integrations.

Disease - October 27, 2023, 2:08 pm

(Oct 27, 2023 02:00 PM)perm Wrote: [\[CITE\]](#) I agree but disagree with your conclusion that it's not a legal consideration. Look at this link how crazy people interact with Wikipedia and why they don't have editors; they are protected by Section 230. But noobs will still threaten to sue every time someone writes something they don't like. <https://www.eff.org/issues/cda230/successes/wikipedia>

If Google or Meta for example, releases lots of packages that have severe errors, and some people get hacked because of poor firewall settings, you bet some noobs will come at them with class-action lawsuits.

I didn't touch the Wikipedia element. I don't disagree with you there. But that's a different thing all together. They're protected by a handful of laws and regulations around that. But at the end of the day it's also just not a part of their business model to curate and edit the content. The whole point of Wikipedia is for it to be the world's encyclopedia, after all.

Google and Meta both do publish loads of packages. You should check out their open source efforts sometime ([here's Google](#) and [here's Facebook](#)). Or just check them out on Github ([Google](#) / [Facebook](#)). Again, open source projects get released under copyleft licenses. And a common theme amongst all of them is that there's no warranty for the user. Aka- if you use it and something goes bad, you cannot go after the package maintainer.

If that wasn't the case there would be no such thing as open source. You'd have individuals, never mind companies, getting sued all of the time. The reason why open source exists and why it's successful is because you can't sue the maintainer like that.

You're also confusing issues again. If you use some open source projects in your codebase and there's a vulnerability, it's you who might be sued by the user of your software. Particularly if you're providing warranty or guarantee on something, or are proven negligent in protecting PII or PCI.

Hackers - December 31, 2023, 9:32 am

(Oct 27, 2023 02:08 PM)Disease Wrote: [\[CITE\]](#) You're also confusing issues again. If you use some open source projects in your codebase and there's a vulnerability, it's you who might be sued by the user of your software. Particularly if you're providing warranty or guarantee on something, or are proven negligent in protecting PII or PCI.

Yeah, a lot of the writers of OS projects have specific license clauses and disclaimers to protect themselves against your use of it. If they're not providing any sort of protection, SLA, support, or bug patch guarantees, then it's really up to the "next" developer (who uses the open source code) to ensure it is safe and secure in their own implementations. Then, the end-user would definitely be suing the app/site/etc they're using, not the original creators of the open source code.

Infinity. - March 4, 2024, 7:39 am

(Oct 27, 2023 08:11 AM)nodes Wrote: [\[CITE\]](#) I'm not talking at all about internal setups or tools. I understand completely what you're saying, it's just I work with a number of enterprise clients and some are publicly traded. We have account reps at a lot of big service providers who are able to tell us exactly what's available, even non-public tools. We still don't get access to their internal tools, but for this I'm specifically talking about how many big companies do not publish public packages on NPM or other package managers even though there is nothing preventing them from being made.

I think that it is a mistake for certain companies to ignore this because ease of development can be a huge reason that people recommend a partnership to a big client. When a package is released, stable, and well-documented, it makes it much easier to produce accurate estimates, plan 1+ year scopes, etc. I think that there is a lot of profit, or open source notoriety that can be gained from releasing packages for certain companies, so that's my goal with that.

Yep, 100%. 'npm install hf' versus installing Axios/fetch/etc and having to manually write out all the API connections, exceptions, tests, etc. Even if a repo codebase can be shared with manual connections to an API, having a stable package release that handles that is a great way to avoid spaghetti code, dev mistakes, and having to rewrite duplicate code across multiple projects. Here's what I understood

simpleApp
-package A
-package B

complexApp
-package A
-package B
-package xyz

You want to club some packages into a single package?

That's too much to maintain(including ci/cd pipelines) to be honest, whenever package upgrade happens you need to upgrade your packages and also need to check if they don't break with existing codebase.

For example, we recently at work upgraded a huge Lambda(typescript) project which was running typescript 3.x to typescript 5.x and it was a horror to begin with.

Upgrading 1 package at a time to its typescript 5 compatible release would at times not work with other existing packages, etc.

Regarding 'npm install hf'. So you will provide API wrapper around HF api with tests so that users don't need to manually install packages and make the API call instead now, users will call your wrapped API instead.

Hackers - March 4, 2024, 7:45 am

(Mar 4, 2024 08:39 AM)Infinity. Wrote: [\[CITE\]](#) Here's what I understood

simpleApp
-package A
-package B

complexApp
-package A
-package B
-package xyz

You want to club some packages into a single package?

That's too much to maintain(including ci/cd pipelines) to be honest, whenever package upgrade happens you need to upgrade your packages and also need to check if they don't break with existing codebase.

For example, we recently at work upgraded a huge Lambda(typescript) project which was running typescript 3.x to typescript 5.x and it was a horror to begin with.

Upgrading 1 package at a time to its typescript 5 compatible release would at times not work with other existing packages, etc.

Regarding 'npm install hf'. So you will provide API wrapper around HF api with tests so that users don't need to manually install packages and make the API call instead now, users will call your wrapped API instead.

for the npm install hf, it would be a package that includes functions for auth & other API routes. it would call direct to the HF API and use your local config / variables for the client id/secret.

I'm just saying in this thread that a lot of companies spend the time to make APIs but don't create a

node module package to use it.

Infinity. - *March 4, 2024, 8:04 am*

(Mar 4, 2024 08:45 AM)nodes Wrote: [\[CITE\]](#) for the npm install hf, it would be a package that includes functions for auth & other API routes. it would call direct to the HF API and use your local config / variables for the client id/secret.

I'm just saying in this thread that a lot of companies spend the time to make APIs but don't create a node module package to use it.

Yup that's true, let me know if you are looking for help, I can contribute.

Hackers - *March 4, 2024, 8:05 am*

(Mar 4, 2024 09:04 AM)Infinity. Wrote: [\[CITE\]](#) Yup that's true, let me know if you are looking for help, I can contribute.

I have a Python codebase & a PHP codebase for HF API but haven't done a JS codebase yet related to HF API. But for other things I'm always interested in making new integrations like this thread's topic - but to be honest, ever since I've been getting deeper into C++ I'm finding it harder to touch anything JS related lol.