

# Forum:[Hack Forums](#)

Thread: [Code obfuscation on GitHub](#)

**macfan** - April 27, 2020, 12:18 pm

Cloned some repo (by importing it; dowl want to be dependant upon some f\* idiots who will (at best) ignore my PR and it will never see a daylight.

What makes me boil is that the devs obfuscated code! Seriously, having contributed to way too many REAL OSS projects, it was FIRST time Ive come across rudeness and arrogance at this scale.....

I would understand commercial project to obfuscate code so noone can "steal" it. OK fine, really. But OSS project, hosted on platform that allows to share code freely? Someone must be an IDIOT. Sorry, cannot find better word, only worse....

**Charger** - April 27, 2020, 2:22 pm

There are 101% zero reasons to obfuscate open source code, especially on a dev branch.

[Nachfolger](#)

link the repo

**Nachfolger** - April 27, 2020, 4:56 pm

I always MD5 hash my opensourced code so that no one else can hack me.

**Joeykh** - April 28, 2020, 5:45 am

Just carefully read the license that comes with the repository.

<https://help.github.com/en/github/creating-cloning-and-archiving-repositories/licensing-a-repository>

**macfan** - April 28, 2020, 7:45 am

Repo Link? Maybe will not link repo as is, but rather specific file (or should I say files)

[Link here](#)

As of sha-suming files its also stupid. Not to mention against OSS idea.

**Charger** - April 28, 2020, 5:27 pm

(Apr 28, 2020 07:45 AM)macfan Wrote: [\[CITE\]](#) Repo Link? Maybe will not link repo as is, but rather specific file (or should I say files)

[Link here](#)

As of sha-suming files its also stupid. Not to mention against OSS idea.  
Modernizer is the only minified non-minified script?

Minified scripts increase speed of script loading.

**Ghost.13** - April 28, 2020, 5:37 pm

Looking at your link, I don't think the code is obfuscated. It's minified, as the .min subextensions specify. Although the result is similar in the context that it's relatively unreadable in one glance,

minification and obfuscation are two different things. While we can never be sure if the files were minified to be obfuscated as well, I can certainly say it was minified primarily for optimization. PrestaShop is an e-commerce platform, something served over the web. We JS developers minify our source code to reduce file size. All those spaces and newlines gone (minification), and variables changed from readable variable names to single character letters (mangling) drastically reduce file size of JS files served over the web. This is mainly used for JS-heavy web applications.

If you want to be sure that JS code is obfuscated, you check [this module](#) for actual JS obfuscation.

**MavenGradle** - April 29, 2020, 1:21 pm

This turned out to be quite the facepalm on your end. You have no clue what you're doing, do you?

First of all, as the others have pointed out, these files are minified versions of their former selves in order to speed up the network delivery to client machines. The reason they include them in their repository is probably because they are JQuery developers that haven't the slightest clue how to work with package managers and content delivery networks. I mean, their development branch and the admin-dev folder does appear to utilize NPM in some areas but they're not doing it right because they're not including their dependencies there and probably have no idea how to properly link all their modules together (or how to properly work with NPM in the first place).

Additionally, ALL of the files in that particular folder on the repository that you linked are very popular JS libraries and themes such as Bootstrap and JQuery. If you're doing anything JavaScript related, you should have at least heard of them. Granted, some of these libraries do seem rather old and outdated, but I guess you can never be surprised by how far back in time a JQuery developer will go.

Allow me to link you to the repositories of the dependencies, for the ones I was able to find: [Bootbox](#)

[Bootstrap lightbox](#)

[Bootstrap modal](#)

[Bootstrap](#)

[Dropzone](#)

[JQuery \(ew\)](#)

[JQuery UI \(also, ew\)](#)

Next time, you might want to do some research before laughing your ass off in regards to open source work. The only thing the repository you linked is "guilty" of is not respecting the original developers by linking to the original content in some cases, possibly breaking some of their license terms. I'm not going to go through the license terms of those dependencies but feel free to do so if you have some extra time on your hands.

Anyway, I do agree with the fact that this repository is a complete mess to traverse and even work with. I don't know why it would receive so many stars because quite frankly, the way their development is handled is quite nasty ([they even recommend working with TortoiseGit in their code contributing documentation and the final section is actually hilarious to read](#)) but then again, most people don't care about the source code, they just want a working product, preferably free, for which

I cannot blame them. I can blame the developers though, so if any PrestaShop devs happen to come across this post, fix your repository and stop working with old and outdated libraries for crying out loud.

**macfan** - April 30, 2020, 6:07 am

@MavenGradle: are you, by any chance, suggesting that jQuery devs are unexperienced newbies?

**MavenGradle** - April 30, 2020, 9:52 am

(Apr 30, 2020 06:07 AM)macfan Wrote: [\[CITE\]](#) @MavenGradle: are you, by any chance, suggesting that jQuery devs are unexperienced newbies?

My comments regarding JQuery developers in a previous post was specifically intended for the developers of PrestaShop.

However, I will say this: JQuery is obsolete and should no longer be considered for modern website development. Vanilla JavaScript has, over time, evolved and has taken care of the features that JQuery "added" and might I say that it has even surpassed it in terms of features. Not to mention that Vanilla JS performs better as well, with the added bonus of not having to specifically load it into your website, which reduces the time it takes to load your website in general (even if it's just a couple of milliseconds). Bloat, that's what JQuery is in this day and age. The only reason to learn JQuery today would be to maintain legacy websites and if that's your cup of tea, by all means go ahead. Any modern JavaScript developer that has even the slightest clue of what they're doing and what JQuery is would tell you the exact same thing.

10 years ago, when JavaScript was missing plenty of features, it would have been a viable library to utilize. Welcome to 2020 though, the year in which you can confidently say that JQuery should be erased the internet. Actually, it could have been removed way sooner than 2020 but people are so slow to pick up on their learning game and tend to stick around with their old and trusted tools, even if they're completely unnecessary. Not to mention the fact that a lot of companies don't tend to update their websites, because of financial reasons. Again, something that is out of the control of the developers, but like I said, JQuery is fine for legacy maintenance only.