



can't resolve rockabilly-loader parsed request is a module webpack sack ov X



About 8 results (0,70 seconds)

Showing results for **can't resolve rockabilly-loader parsed request is a module webpack *stack* overflow**

Search instead for **can't resolve rockabilly-loader parsed request is a module webpack sack overflow**

<https://stackoverflow.com> › questions › module-not-fou...

Module not found: Error: Can't resolve 'fs' in - Stack Overflow

23 Jul 2019 — **webpack**: (config, { buildId, dev, isServer, defaultLoaders, **webpack** }) => {
config.node = { fs: 'empty' } return config },. In other cases, please refer ...

9 answers · 57 votes: The error is because of angular-cli does not support modules in node lik...

[More results from stackoverflow.com](#)

Missing: ~~rockabilly-~~ | Must include: **rockabilly-**

Edit Manage Stats



Ingo Steinke

Posted on Oct 22, 2021 • Updated on Aug 29



10



2

I enjoy life-long learning, but...

[#webdev](#) [#watercooler](#) [#devjournal](#) [#rant](#)

... there are some aspects of working as a web developer that do not spark joy in my heart. After collecting some notes over time, I have pasted them into this dev rant and added some links, screenshots, and clarification.

Is Learning a Waste of Life?

I enjoy life-long learning, especially of frontend technology like HTML, CSS, UX, UI, also backend stuff like SQL, Doctrine, and Symfony. So many things have improved (even JavaScript and PHP), but then there is stuff that I could do without, like spending days configuring Webpack and its plugins only because somebody else decided that every project needs to use Webpack (replace Webpack with any other tool or task you could have done without or delegate it to somebody else).

Delete Bookmark

Things that I could do without

This rant is basically a collection of some stuff that I had written down to get it out of my mind. Finally decided to publish and share, so maybe it's fun to read, and maybe it is good for you to know that you are not the only one getting annoyed and confused by misleading error messages and the like. So here we go:

Breaking Changes

Adapting boilerplate configuration code for Webpack, React, or any other tool, to reflect the latest breaking changes.

Tasks like this may mean front-end web development to some people. But to me, it means nothing but wasting my time with menial tasks that could have been avoided if developers would actually care not to break things.

```
No fix available
node_modules/vorpal/node_modules/inquirer/node_modules/lodash
  inquirer <=0.11.4
    Depends on vulnerable versions of lodash
node_modules/vorpal/node_modules/inquirer
  vorpal *
    Depends on vulnerable versions of inquirer
node_modules/vorpal
  @frctl/fractal 0.8.0 - 1.5.11
    Depends on vulnerable versions of vorpal
node_modules/@frctl/fractal
  fractal-webpack-plugin *
    Depends on vulnerable versions of @frctl/fractal
node_modules/fractal-webpack-plugin

23 vulnerabilities (13 moderate, 5 high, 5 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.
```

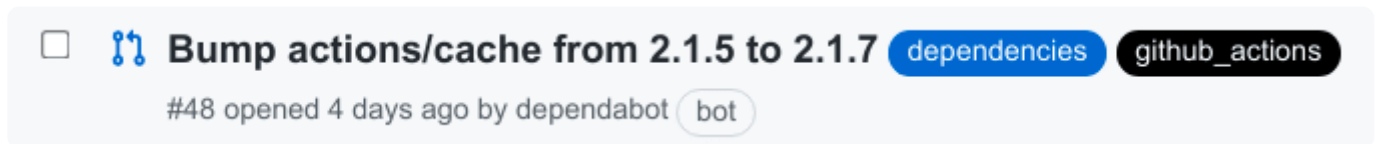
If it ain't broke, don't fix it

Apart from regular upgrades to node packages, which break other packages' peer dependencies so that those other packages need another upgrade, which breaks another package ... apart from that, we can check for security vulnerabilities automatically using `npm audit`, which is, in theory, a good idea, despite occasional irrelevant alerts and false positives. But either there is "no fix available" or if there is, chances are, that `npm audit fix` will upgrade (or even downgrade) a dependency that breaks something else.

♥ Stylelint: TypeError: [configExtends].flat is not a function

Broken Dependencies

Don't even bother checking the error logs and vulnerability details to try and fix the problems yourself. You will find out that half of your software has already been deprecated in favour of another totally different technology, which is either unavailable due to a peer dependency of an indispensable module, or else it will be completely incompatible, poorly documented and still in beta.



Hidden Assumptions

Beta or not, even common tools that are supposed to be stable, maintained, and popular, sometimes lack the ability and documentation to make them "just work" on my own machine.

Every tutorial seems to make at least one assumption that does not match my current situation when trying to use any popular tool exactly like advised in their official documentation.

"I tell you, it's easy!"

Annoyed also by tutorials trying to tell me that something is easy, all I need to know, or supposed to make me have fun.

I am just trying to do my work, okay?

Okay, we'll send you a GIF.



Silly!

Annoyed to see animated meme images of some seemingly popular Hollywood actor, when all I wanted was a solution to a technical problem. Having to read npm idiosyncrasies like "sill" or silly slack phrases like "Sweet potato! Here's a tractor" or "more holy moly!"



**You're all read.
Here's a pony.**

Does that make sense?



Useless Knowledge

[Yagni](#) is a statement that some capability we presume our software needs in the future should not be built now because "you aren't gonna need it". The same could be said for many facts that I learned the hard way, without any relevance for future use.

Researching and learning anything could always be considered a waste of time in the perspective of being outdated and irrelevant in the future, but then at least the learning process has an intrinsic benefit.

Or at least it should have, while in practice, we often spend hours, days, even months and years, to research specialized knowledge that does not provide any benefit outside its limited use case.

I have emphasized the positive aspects in another blog post ([Productive Procrastination vs. Apparent Productivity](#)).



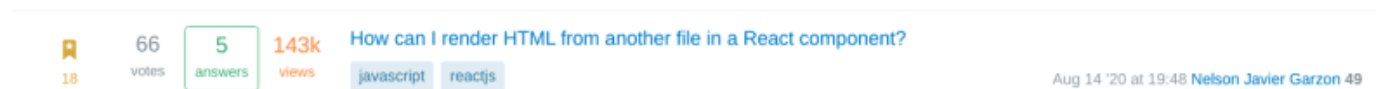
Productive Procrastination vs. Apparent Productivity

Ingo Steinke • Mar 21 '22

#productivity #creativity #watercooler #motivation

But let's focus on the unproductive and annoying aspects now.

I am not talking about jQuery, Flash, or BASIC. All of them gave me insights and experience that I don't want to miss, seriously! But looking back at my StackOverflow bookmarks set in the past years, I deleted numerous links to discussions about Highcharts options, Internet Explorer idiosyncrasies, Docker glitches on MacOS, Network Manager malfunctions on Ubuntu, line-endings issues in git bash on Windows, ReactDOM peculiarities, and third-party services' API problems.



When I think of all the good time that's been wasted!

Overly verbose error messages are another potential time sink.

We could also call them...

Stack Trace Novels

Error messages containing several pages of verbose stack trace details that still omit the actual file name or line number that would help you to trace the error back to your source code.

If line numbers are included, they are either lines in the source code of a third party framework component, or line 1 of your compiled application bundle. I even got errors in line 19 of a component that only had 11 lines in my source code.

I'm seeing this issue as well. Here's the stack trace:

```
Uncaught TypeError: Cannot read property 'props' of undefined
    ReactElement.cloneElement @ ReactElement.js:235
    ReactElementValidator.cloneElement @ ReactElementValidator.js:451
    render @ app.jsx:95
    ReactCompositeComponentMixin._renderValidatedComponentWithoutOwnerOrContext @ ReactCompositeComponent.js:789
    ReactCompositeComponentMixin._renderValidatedComponent @ ReactCompositeComponent.js:816
    ReactPerf.measure.wrapper @ ReactPerf.js:70
    ReactCompositeComponentMixin.mountComponent @ ReactCompositeComponent.js:237
    ReactPerf.measure.wrapper @ ReactPerf.js:70
    ReactReconciler.mountComponent @ ReactReconciler.js:38
    ReactCompositeComponentMixin.mountComponent @ ReactCompositeComponent.js:247
    ReactPerf.measure.wrapper @ ReactPerf.js:70
    ReactReconciler.mountComponent @ ReactReconciler.js:38
    ReactCompositeComponentMixin.mountComponent @ ReactCompositeComponent.js:247
    ReactPerf.measure.wrapper @ ReactPerf.js:70
    ReactReconciler.mountComponent @ ReactReconciler.js:38
    mountComponentIntoNode @ ReactMount.js:248
    Mixin.perform @ Transaction.js:134
    batchedMountComponentIntoNode @ ReactMount.js:269
    Mixin.perform @ Transaction.js:134
    ReactDefaultBatchingStrategy.batchedUpdates @ ReactDefaultBatchingStrategy.js:66
    batchedUpdates @ ReactUpdates.js:118
    ReactMount._renderNewRootComponent @ ReactMount.js:484
    ReactPerf.measure.wrapper @ ReactPerf.js:70
    ReactMount.render @ ReactMount.js:493
    ReactPerf.measure.wrapper @ ReactPerf.js:70
    (anonymous function) @ app.jsx:31
    S15.mover/config/config @ app.jsx:33
    s @ _prelude.js:1
    e @ _prelude.js:1
    (anonymous function) @ _prelude.js:1
```

I think it might concern `react-router` because it starts off from `app.jsx` :

Search Overflow

[Googling the error message](#), finding either unrelated or unanswered questions on StackOverflow, and if someone did have the exact same error message, surely their question had been erroneously marked as a duplicate or "downvoted" and closed for any other obscure reason.

Searching Without Matches 🔥

It looks like there aren't many great matches for your search

Tip: Try using words that might appear on the page you're looking for. For example, "cake recipes" instead of "how to make a cake."

In other cases: not a single matching entry, and Google stating the obvious: that there are not many great search results. You don't say! 🤔

Missing, must Include

Alternatively getting a lot of search results, but mostly unrelated, annotated by small blue links with a strike through one of my search terms, stating "missing / must include". These links never provided any single more relevant result when clicked.



can't resolve rockabilly-loader parsed request is a module webpack sack ov X



About 8 results (0,70 seconds)

Showing results for [can't resolve rockabilly-loader parsed request is a module webpack **stack** overflow](#)

Search instead for [can't resolve rockabilly-loader parsed request is a module webpack sack overflow](#)

<https://stackoverflow.com> › questions › module-not-fou... ⋮

Module not found: Error: Can't resolve 'fs' in - Stack Overflow

23 Jul 2019 — **webpack**: (config, { buildId, dev, isServer, defaultLoaders, **webpack** }) => {
config.node = { fs: 'empty' } return config },.. In other cases, please refer ...

9 answers · 57 votes: The error is because of angular-cli does not support modules in node lik...

[More results from stackoverflow.com](#)

Missing: rockabilly- | Must include: [rockabilly-](#)

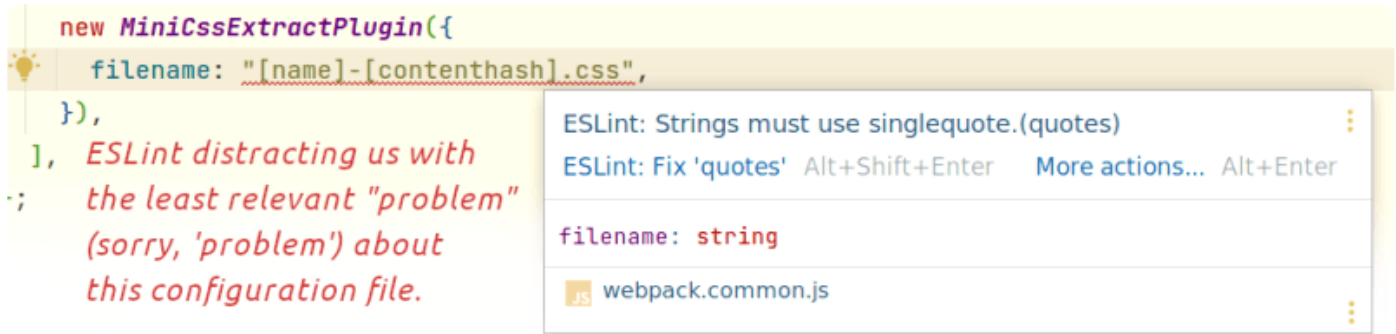
SEO Spam Experts

Using search engines to find solutions, we all have stumbled upon those beginner-friendly tutorial blog posts with their spammy click-bait headlines featuring outdated information, like **"7 JavaScript Idioms that every developer MUST know in 2023"**, featuring a list of six code excerpts, five of which are either outdated, insecure or not working at all, originally written in 2016, edited year after year just to increment the date to confuse novice developers who just started to write their first lines of code.

Those content creators are experts when it comes to search engine optimization and monetization, using blinking banners and links to dubious third party vendors or affiliate partners. You will see them on their own websites, on YouTube, and - unfortunately - even here on DEV. I don't know why advertisers even allow those people to profit from the monetization, but then again, low-quality content and low-quality ads are a good match, and at least they help me to recognize bullshit more easily. Sorry for the rant, but I think it would help to improve the quality and reputation of development if that kind of content would be deleted forever.

Linting the Wrong Code

All of this only to find out later, after hours of frustrating research, that the actual error was a typo in a completely unrelated file never mentioned by any error message or linter warning.



"Strings must use singlequote": ESLint distracting us with the least relevant "problem" (sorry, 'problem') about this configuration file.

Another example of well-meant rules that don't help to achieve the intended goal is StackOverflow, also known as the online museum of jQuery code snippets.

Not Allowed to Fix StackOverflow

Trying to fix the same typo in the StackOverflow example where it came from, but on submitting an edit, getting the error message from StackOverflow:

Edit Summary

fix typo in Name

Save edits Cancel

Edits must be at least 6 characters; is there something else to improve in this post?

"Edits must be at least 6 characters; is there something else to improve in this post?"

[Change this behavior to allow for spelling corrections and the like](#) has been requested on StackExchange Meta for at least 10 years. Not the only reason to rant about StackOverflow, but this one seems really silly.

As silly as the privilege system not allowing me to edit anything apart from my own posts, thus all I can do is suggest an edit to be put into an edit review queue which is always full, where it will be an annoyance to overworked senior reviewers with a higher reputation, wasting their time approving typographic and formatting fixes instead of being able to focus on conceptual changes.

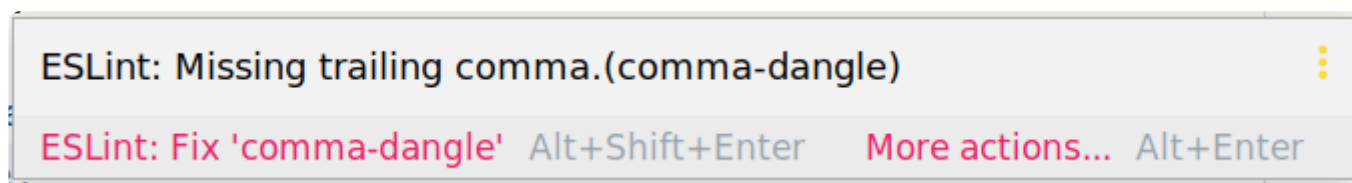
So, like in a bureaucratic state's public service agency or a toxic corporate workplace, they have managed to create a system where different actors get annoyed for

completely different reasons and everybody's unhappy at the end of the day. At least, this system has been discussed and documented a thousand times, so we can actually learn about every single detail making it toxic and dysfunctional.

Fair play, maybe it's still a more efficient and effective system than most others, and we just stopped to question the more dysfunctional ones or never even expected them to work properly in the first place. Like politics, train service or trying to establish a mobile internet connection in Germany.

AirBnB's Coding Style

Meanwhile, linters keep nagging about missing or superfluous braces, spaces or indentation, again instead of providing any insight about the actual problems in my code. Even more so, if a senior team member or a company coding style guide insisted on using [AirBnB's JavaScript Style Guide](#). Their website looks neat, and so did most of the places I ever booked over their platform, but their JavaScript style guide still sucks. Apart from my personal taste, I tried to provide some reasonable arguments for my dislike.



Dangling Characters

Adding dangling commas at the end of any line where you do not need to. Okay, this is even valid ECMAScript, at least, so no need to transpile for the linter's sake. But I did not find it helpful to achieve the intended benefit of getting less verbose `git diff` to review, because adherence to indentation rules makes everyone reformat twenty lines of code around even a single character fix on one line anyway.

10:25	error	Missing space before function parentheses	space-before-function-paren
13:1	error	This line has a length of 128. Maximum allowed is 100	max-len
14:17	error	Missing trailing comma	comma-dangle
15:6	error	Trailing spaces not allowed	no-trailing-spaces
18:20	error	Newline required at end of file but not found	eol-last
18:20	error	Missing semicolon	semi

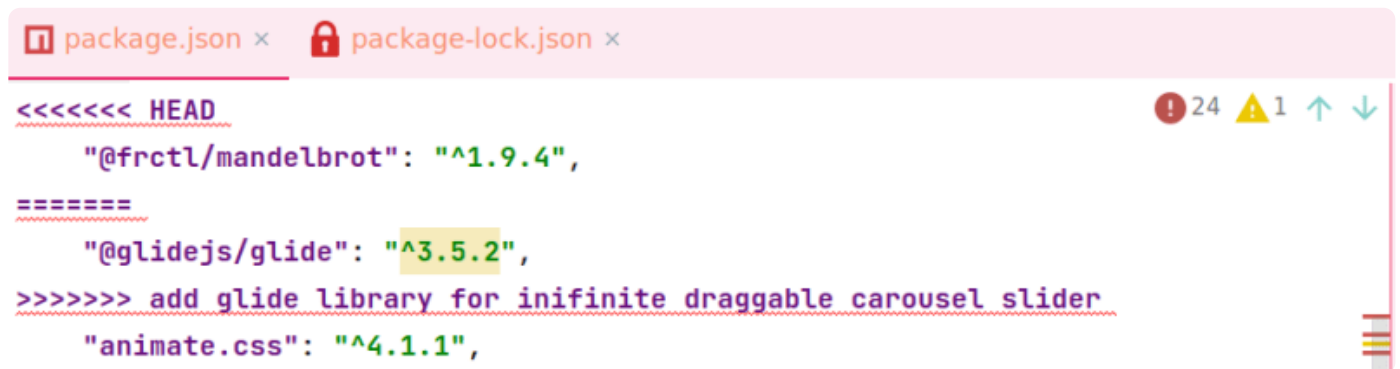
Having to add trailing commas mostly everywhere just moves inconsistency to `package.json` and any other `json` file where there are still some parts of the code that will make your code invalid. Now why won't the ubiquitous [Babel](#) handle that as well?

Popular Double Quotes

Most code examples found in official documentation seem to use double quotes for strings. AirBnB doesn't. So after pasting documentation examples into your own code, you have to "fix" the "errors", sorry: 'fix' the 'errors' until you see no more strings must use single-quote.

But even the best code style rules, [.editorconfig](#) and [trunk-based development](#) will not save you from the ubiquitous

Merge "Conflicts"



```
<<<<<< HEAD
    "@frctl/mandelbrot": "^1.9.4",
=====
    "@glidejs/glide": "^3.5.2",
>>>>>> add glide library for infinite draggable carousel slider
    "animate.css": "^4.1.1",
```

Not really an issue in most cases, but still another waste of time and concentration, when every now and then, using more than one branch, small edits, merge, and rebase, git diff complains about "conflicts" that are obvious to solve for a human but still not for a machine, despite the hype about the age of AI, [coding co-pilots](#) and the like.

Expected Mess

It still happens, that you copy a well-formatted code snippet and paste it into your editor / IDE where the original format gets messed up to unreadability, does not get fixed, but instead linted and underlined with helpful annotations like "Expected indentation of 8 spaces but found 4".

Recommended Mess

All of this mess gets more likely, the more recommended tools you try to stick together, especially if any part of the setup or boilerplate code has not been updated during the past eleven months.

Case Closed

Seen the same pattern in GitHub issue, but also in Google's Help Center: there is one single, unhelpful, answer for a problem that keeps occurring and which is hard to google, and the thread is already closed? "This question is locked and replying has been disabled." Also known as "Can't Reproduce, case closed."

Selling Stolen Information

Last, but not least: I am really upset about companies trying to make me pay for information that is, or should be, freely available. Don't get me wrong: I am happy to pay for expertise and good education. Paid courses on Udemy and, even more so, [FrontendMasters](#) have provided a source of valuable knowledge, and I will surely take more courses when I find time to do so.

On the other hand, there used to be Experts Exchange (always remember EE when you're feeling bad about StackOverflow) and now there is medium.com and various code grepper pages, full of articles and code snippets, many of them dubious, mediocre, and badly written, trying to recycle open-source knowledge to sell subscriptions or making money by displaying annoying ads all over their page.

Your search did not match any documents.

Your search did not match any documents.

Suggestions:

- Make sure that all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.



Top comments (3)



Mike Harrison • Apr 8 '22 



Lol at least all the eslint things you're complaining about can be auto fixed so it's not a big deal. I haven't noticed double quotes in documentation though, maybe we hang out in different documentation but mostly I've seen single quotes, and most likely cause everyone is using the same Airbnb config.



Mike Harrison • Apr 8 '22 



I feel you on a lot of the other stuff though...especially npm upgrades. Such a nightmare.



Ingo Steinke  • Nov 27 '21 



After all the wasted time fighting against and together with ESLint, Google, and StackOverflow, I have wasted even more time by updating this article! Added several screenshots and collages, fixed spelling errors, and added some more details and issues. Have fun! Also, I am curious about programming^H^H^H^H --- no, I mean, I am curious about your stories, rants, and funny failures in the comments.

[Code of Conduct](#) • [Report abuse](#)