

Why were the new JSLint errors "use spaces, not tabs" and "unsafe character" introduced?

Asked 12 years ago Modified 9 years, 1 month ago Viewed 35k times



60



I have been validating my JavaScript using JSLint for about 2 years now and once in a while there are rules that change. In general when JSLint introduces a new rule, there is a checkbox to ignore this rule when parsing, or if you choose to not ignore it then to make your code compliant to it.

As I was running my JSLint validation today, however, I run into these two new errors:

Use spaces, not tabs.

This is not the "mixing of tabs and spaces" error. I am using only tabs. This is a recently modified version of "mixing of tabs and spaces" which now disallows tabs in general.

And:

Unsafe character.

```
*/
```

Unsafe character.

```
_const: {
```

There are no new options to ignore. I cannot understand what is unsafe about closing a block comment, why it considers `_const: {` as unsafe when I have `nomen: true`, (dangling `_` in identifiers) or why should I be suddenly switching from spaces to tabs, when I still have the configuration about indentation of 4 spaces being a tab.

Does anyone have an idea why those were introduced to at least how to make JSLint ignore these new rules?

Update: The `Messy White Space` option works around the issue but it would cause other unexpected behavior:

```
if (condition) {  
  //      ^-- there is a space but it won't indicate an error
```

Share

Improve this question

Follow

edited Jun 20, 2020 at 9:12



Community Bot

1 • 1

asked Dec 17, 2012 at 11:23



Konstantin Dinev

34.9k • 14 • 78 • 102

Please, post your linter config. – oleg Dec 17, 2012 at 11:26

- 6 `*/` is considered unsafe in JSLint because the creator of JSLint believes it is safer to use `//` comments on each line rather than block comments, as there is a chance of human error were you to accidentally put `*/` inside your comment and close the block too early. Personally, I ignore it. – Dawn Dec 17, 2012 at 11:37

@Dawn I am not getting it only at that line, I will update with another where it makes even less sense. – Konstantin Dinev Dec 17, 2012 at 11:38

It was to do with the regular expressions and `/**/` comment it could be acedentaly tripped by some regular expression like `*/`. So Douglas Crockford sad that you should use `//` single quotes. They commented out whole line(between line brakes), or if you put them on a line whit some code everything from `//` to the end of the line. – IGRACH Nov 16, 2013 at 6:55

- 2 If the line breaks are accidentally removed from your code, then `//` will comment out the rest of the file, while `/* */` won't, so to say that the first is safer is a bit naive. – Chris Middleton Nov 24, 2014 at 14:56

4 Answers

Sorted by: Highest score (default)



Well it looks like Douglas Crockford just made a whole lot more people switch to JSHint. Have a look at [this commit](#).

77



The "Mixed spaces and tabs" error has been removed, and a new "Use spaces, not tabs" error has been added in its place. Aside from that, there's one tiny change in that diff that shows the cause of this. The following line (comment added):



```
at = source_row.search(/\t/);
//                      ^ Space
```



has been replaced with this:

```
at = source_row.search(/\t/);
//                      ^ No space!
```

Following that search there's an `if` statement. If the condition evaluates to `true`, the "Use spaces, not tabs" warning is issued. Here's that statement:

```
if (at >= 0) {  
    warn_at('use_spaces', line, at + 1);  
}
```

I *hope* that this is just a little oversight by Crockford. As you can see, JSLint is now going to raise this warning if you use a tab character *anywhere*. Unfortunately, his commit messages are completely useless, and the documentation doesn't appear to have been updated, so I can't do anything other than speculate as to the reasons behind this change.

I suggest you abandon JSLint and switch to [JSHint](#) right now.

Share Improve this answer Follow

answered Dec 17, 2012 at 13:12



[James Allardice](#)

166k ● 22 ● 334 ● 315

Thanks for investigating this. Switching from JSLint to another validator is kind of not an east thing for me. I have an entire infrastructure built to run automated validations against JSLint... – [Konstantin Dinev](#) Dec 17, 2012 at 13:38

@KonstantinD-Infragistics - JSHint started life as a fork of JSLint. It should be pretty easy to swap them over. Where there's a tool for JSLint there is usually one for JSHint too (e.g. both have Node.js wrappers that allow you to run them from the terminal). Alternatively, are you able to control the version of JSLint in use? – [James Allardice](#) Dec 17, 2012 at 13:40

Yes my Node.js code does not download an updated version and my automated execution would not show these errors until I update the jslint file, however when you manually want to check for errors you can no longer do that on jslint.com. – [Konstantin Dinev](#) Dec 17, 2012 at 13:49

112 With all due respect, fu** spaces. – [Langdon](#) Apr 29, 2013 at 16:16

2 Yeah, it's ridiculous that a validator has to point the usage of tabs in each and every line of your code. It distracts you from the real problems in your code. Plus, what's so wrong about tabs?! – [OMA](#) Oct 7, 2015 at 9:53



You can suppress the error by clicking the "messy white space" option.

13

Share Improve this answer Follow

answered Dec 31, 2012 at 16:05



[mathius1](#)

1,391 ● 11 ● 18





I actually found a flaw in this workaround, but it's still the only thing available so far...

– [Konstantin Dinev](#) Jan 3, 2013 at 9:20

What is the the flaw, @KonstantinD-Infragistics ? – [Belladonna](#) Mar 12, 2013 at 14:41

- 6 I have explained it in the last portion of my question. This way trailing spaces are ignored and they shouldn't be, – [Konstantin Dinev](#) Mar 12, 2013 at 15:05



10



To answer *why* JSLint now gives an error about tabs, <http://www.jslint.com/help.html> gives this justification:

Tabs and spaces should not be mixed. We should pick just one in order to avoid the problems that come from having both. Personal preference is an extremely unreliable criteria. Neither offers a powerful advantage over the other. Fifty years ago, tab had the advantage of consuming less memory, but Moore's Law has eliminated that advantage. Space has one clear advantage over tab: there is no reliable standard for how many spaces a tab represents, but it is universally accepted that a space occupies a space. So use spaces. You can edit with tabs if you must, but make sure it is spaces again before you commit. Maybe someday we will finally get a universal standard for tabs, but until that day comes, the better choice is spaces.

Essentially he wants everybody to come to a consensus on whether to use tabs or spaces to prevent them from ever being mixed. He's decided that the consistency of the width of a space makes it the superior choice, so we should all use that. Clearly some people will disagree with this line of thinking (myself included), but that is the reason why JSLint throws that error.

Share Improve this answer Follow

answered Nov 4, 2015 at 19:08



[Kat](#)

239 ● 2 ● 12

- 25 ...that makes no sense. Because of the defined width that's exactly why spaces WON'T be consistent...because some people will use 2 spaces, some will use 4, etc. So some files indents will be twice as large as others. With tabs it's simply 1 tab == the indent, and your editor with your settings determines how far that indent is viewed as being...and every file using tabs will show the exact same for you. Tab is the perfect placeholder for saying "it's 1 level of indent...however far you want that to mean for your editor", and every file that uses that will have indents looking the exact same. – [Jimbo Jonny](#) Jan 24, 2016 at 3:49

Depending on your editor/IDE you can adjust how TAB works.



For instance, I use Sublime Text. Near the bottom right corner there is a Tab Size : 4.

6

I clicked on it and set it 'Indent Using Spaces'.



This updated all my Tabs to use spaces and JSLint errors disappeared. I try to use as few options as possible with JSLint as I want my code to be well structured.



I also use JSFormat, which will tab based on my editors settings, so whenever I'm done I run my JSFormat, then JSLint. No errors = happy boy!



Hope it helps.

Share Improve this answer Follow

answered Sep 20, 2013 at 17:46



[TyMayn](#)

1,936 ● 2 ● 18 ● 23

Also, for vim, just add this to your .vimrc and never worry about it again: `set tabstop=4 shiftwidth=4 expandtab` – [user1108579](#) Sep 21, 2013 at 12:41

-
- 18 The issue with most editor's option of *indent using spaces* is that you press `Tab` once, but have to press `Backspace` four times. There is no editor, that i'm aware of, that handles tabs-as-spaces correctly as a single entity. – [Ian Boyd](#) Feb 13, 2014 at 15:24

To combat this, many editors have their own indent and unindent hotkeys. For instance, with sublime text `cmd+]` will indent, while `cmd+[` will unindent. Helps when you need to clean up someone else's slop. – [TyMayn](#) Sep 9, 2014 at 4:06

-
- 2 But the question is not about the workaround. The question is, what's wrong with tabs?! Why does this merit a mention in each and every line of your code that contains a tab! It distracts you from the actual problems in your code!! In fact, I think tabs are better because it's easier to have the same number of tabs than spaces (less actual characters) and the files take less space (a tab character takes 1 byte instead of 4), though this is not so important. – [OMA](#) Oct 7, 2015 at 9:50
-