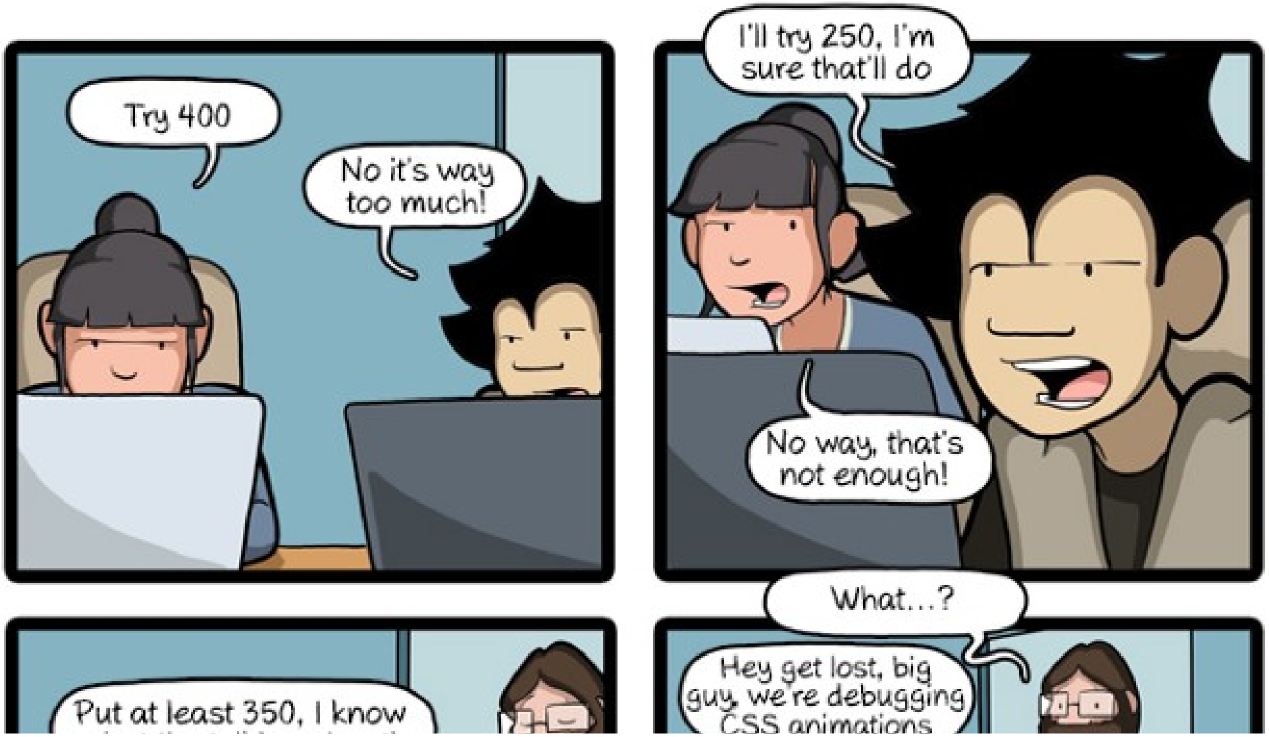
Debugging “just my code dev tools.

Jeffry Steven Lenis Follow
Mar 5, 2018 · 2 min read


Try 400
No it's way too much!
I'll try 250, I'm sure that'll do
No way, that's not enough!
What...?
Put at least 350, I know
Hey get lost, big guy, we're debugging CSS animations

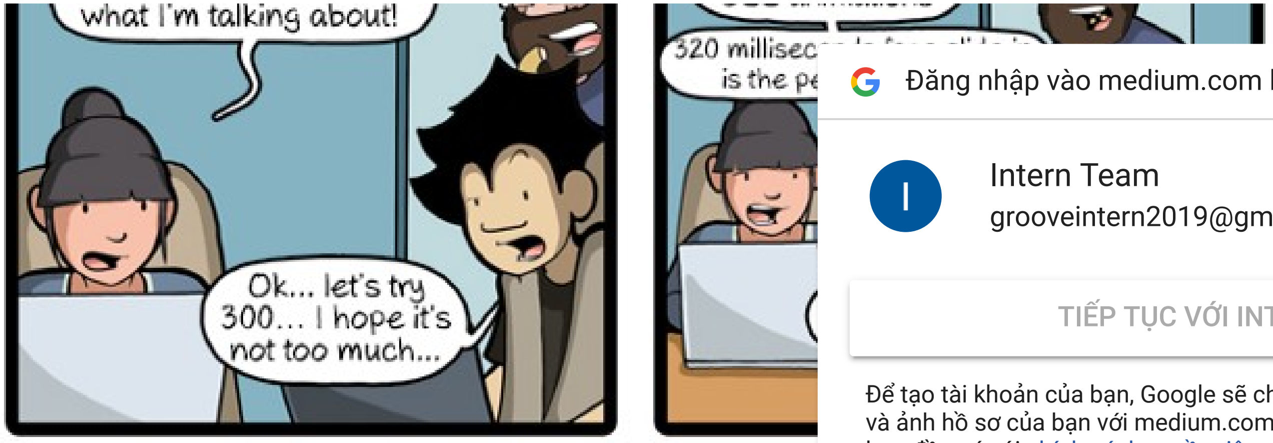


Image by CommitStrip

All Angular applications are now bundled and compiled (that’s so cool), our application is full of npm modules, libraries, etc.. But, we only focus on the little amount of files under the *src* folder. At the end of the day, we just have our **.ts* and **css* files, that is what we build.

What happen if we have an error, or we want to debug a function or component suspicious behaviour?. Commonly we put some console.log’s on the code and “debug” visually the different messages we get on the console. But, maybe these behaviours are difficult to follow and, if we have an error, that error could be difficult to diagnose (specially now, on the backendless/Firebase/BaaS era).

> Errors can (will) happen, every time you write some new computer code

Debugger statement is really good to introduce a
your code efectively. But remember, our Angular
instance with Webpack. Then, when you put tha
to debug, almost in every case you will jump stac
framework/library you use (in this case it’s Angu

We don’t want to travel to the underworld (npm_modules folder), we **only** want to debug **our function**. Good news!, in Chrome developer tools we have cool feature called *Blackboxing*. Blackboxing gives developers the possibility to denote a library code so that the debugger can route around it. When you blackbox a source file, the debugger will not jump into that file when stepping through code you are debugging.

Blackbox JavaScript Source Files - Google Chrome

Like everyone, you get bugs in your application code. You start debugging, but when you step through your code…

developer.chrome.com

It’s really easy to use, open the developer tools and simply right-click in the
file that you want to blackbox; but on bundled ap
apps, this might not be enough. Your function co
framework function located in other file and this
another function maybe in another file and a libr
converting the blackboxing task in an endless cy
can make a pattern of files which must be blackb
regex!).

Into my recent projects, I started to use the next regex and works perfectly on Angular applications.

```
^(?!.*\map$)(?!.*\.ts$)(?!.*\.bundle\.js)(?!.*\.dll\.js).*
```

If you have improvements or another beautiful regex, please tell us :).

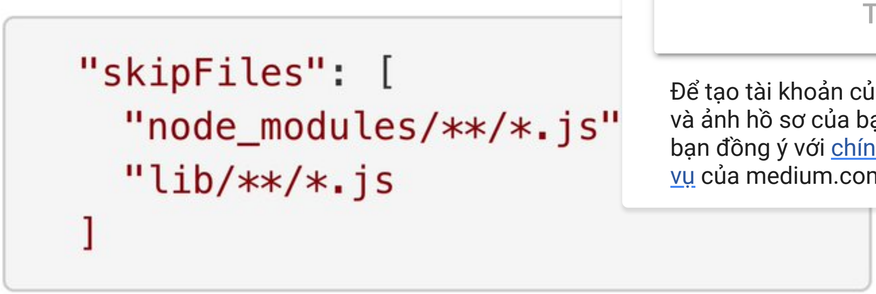
. . .

Bonus:

On VS Code, we can do this, for debugging.

Kenneth Auchenberg
@auchenberg

Our Node and Chrome debuggers for VS @c
"Just My Code" aka blackboxing while debugg
code.visualstudio.com/updates/v1_8#_…



```
"skipFiles": [
  "node_modules/**/*.js"
  "lib/**/*.js
]
```

31 5:56 AM - Dec 15, 2016

See Kenneth Auchenberg's other Tweets

Thanks to Erika Suárez Valencia.

JavaScript Angular Debugging Frontend Blackboxing