Can everyone see your JavaScript code?

Did you ever debug your (compiled) JavaScript code in the browser dev tools?

In case you didn't know - you can do that, for example with the Chrome Developer Tools .

Whilst this is very useful, it's not limited to your own pages. You can inspect **any** JavaScript code on **any** webpage this way. Of course not the uncompiled version (in case you were writing your app with a framework like Angular or a library like React) but still the JS code that holds all your logic.

Doesn't this pose a huge security issue? Because if you can do that, **everyone** is able to do that.

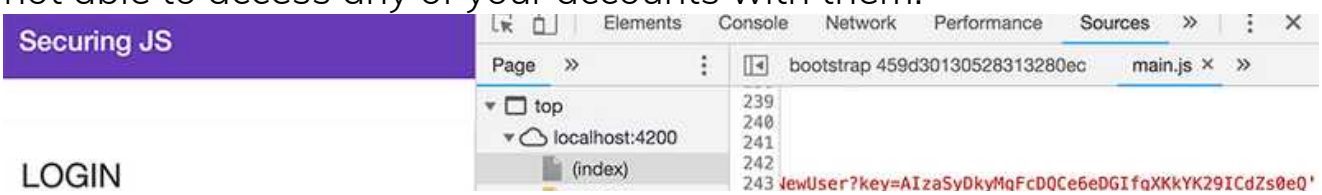
This is not a problem!

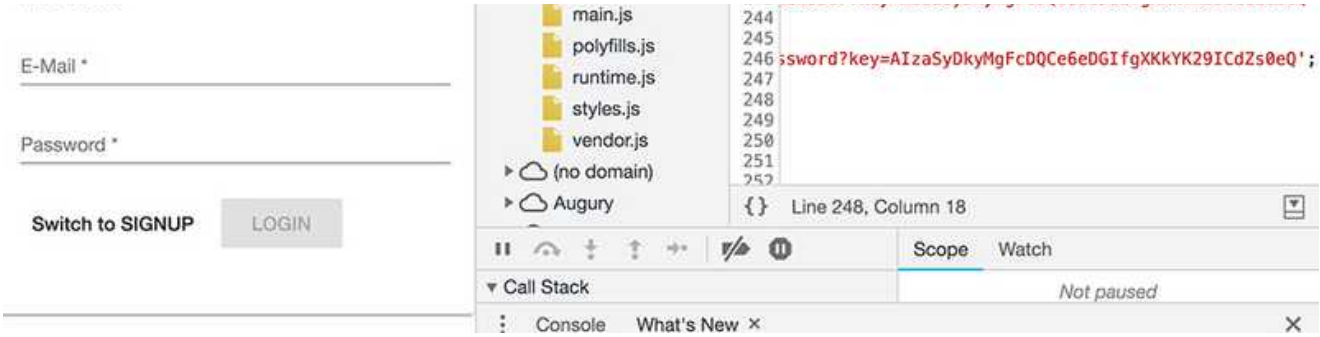
No worries, you're not in danger because of that. At least not if you think about what you put into your frontend-facing JavaScript code.

You certainly shouldn't put your account usernames or passwords in there. The same goes for any other confidential information.

> Want more information on this? Check out the video which you find at the top of this page!

Things like API keys can generally go in there though since people are not able to access any of your accounts with them.





You might want to control access to an API from within your API dashboard - or, if you're the creator of the API, in the code you wrote.

You can set up IP or domain whitelists for example. This would allow you to expose your API key in your frontend JavaScript code (and you typically need it there) and still control which pages are able to use it. That ensures that other people can "steal" your API key but that it's pretty worthless to them.

What should NOT go into your JavaScript code?

So what should **not go** into your JavaScript code?

Anything which gives other users access to any of your accounts. Any customer data (hardcoding customer data is never a great idea by the way...) and in general: Any confidential data.

That's also the reason why you'll never directly connect to a database from your frontend JavaScript code. And Firebase isn't a database in case you're wondering. It's a backend service which you still access through an API (or a SDK that accesses it for you).

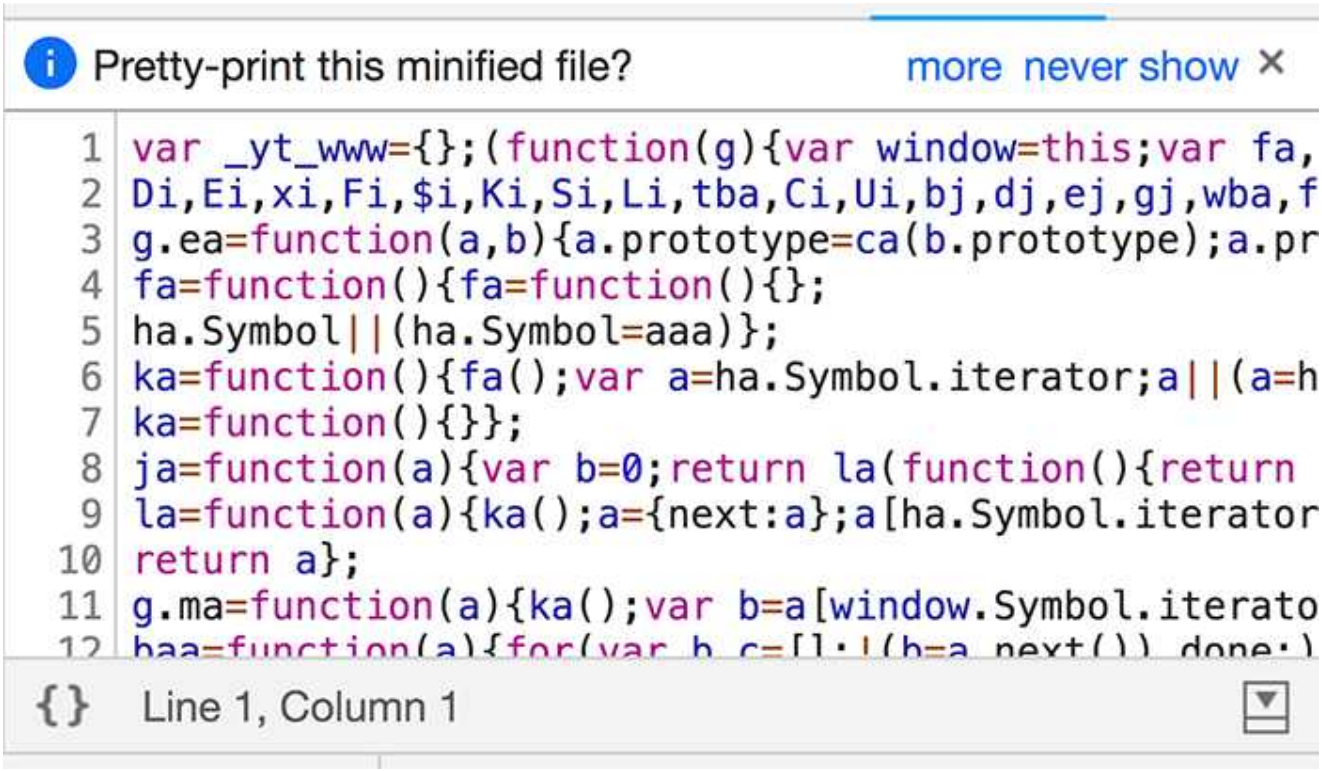
Storing any database credentials or query code directly in your frontend JavaScript code would be a HUGE security issue! Any user who reads it could start accessing your database or manipulate your queries.

> Summary: Don't put any confidential data - e.g. account information or database queries - into your frontend JavaScript code!

What about minification - does that not help?

To conclude this article (and video) - which role plays minification?

It makes your code look like this:



This code looks very unreadable, doesn't it?

Well, it does but it still is. The chrome dev tools even help you transform it!

You can click the `{}` symbol at the bottom of the image to convert it to a more readable version.

Variable and function names will still be shortened but API keys and similar things will be relatively easy to spot.

Minification is NOT a security mechanism!

It only serves one purpose: Decrease the size of your JavaScript code! It's a performance enhancement, not meant to secure anything.

Instead, keep the previous hints in mind and secure your API keys + avoid putting confidential information into your frontend JavaScript code.