

01.

# Developers

Everyday frontend development  
according to 4,500 experts



**Patryk Mamczur**  
Report's Editor in Chief

**” With 4,500 folks filling in the survey, the State of Frontend 2020 is the biggest report out there focused solely on frontend development.**

When we started thinking about the State of Frontend 2020 report, I had one goal in mind: to find out what the everyday job of a frontend developer looks like. Considering that there are thousands of frontend devs around the world, that goal seemed pretty unrealistic. But hey man – once again, the frontend community surprised us all!

In just a few weeks, exactly 4,500 frontend developers took part in our State of Frontend survey! I know the number first hand because I was literally sitting with my finger on a button, observing the growing numbers and waiting for the perfect moment to close the survey. I don't know about you but for me the number is pretty perfect – as with 4,500 folks filling in the survey, the State of Frontend 2020 is the biggest report out there focused solely on frontend development.

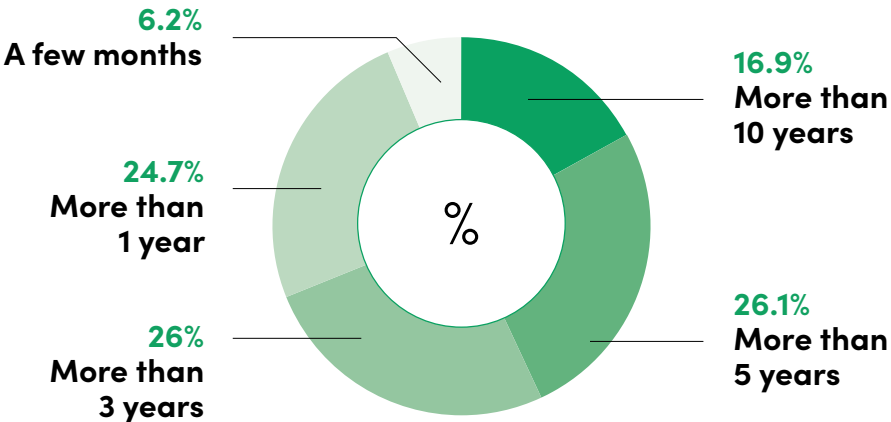
All in all, we ended up with over four thousand frontend devs telling us about their everyday job, the frameworks that they use, the frameworks that they would like to use (but, for example, the boss won't let them) and about their

thoughts on the recent frontend development trends. The results of the survey are awesome – some of them surprising, many of them inspiring, all of them showing how the everyday frontend development looks.

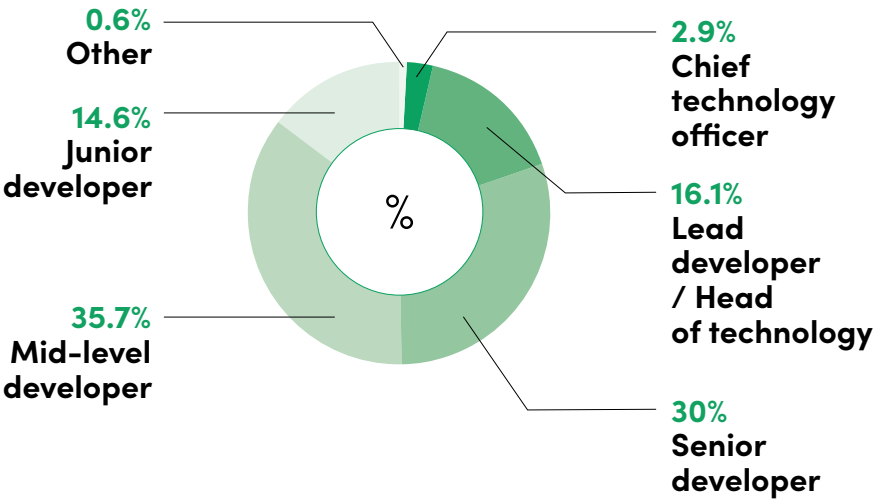
To make all this data more accessible, we invited leading software development authorities to comment on the results. With their incredible know-how and big-picture perspective, they made the State of Frontend 2020 report what it is – the most up-to-date source of knowledge on the modern frontend development.

So, I advise you to stop reading my brag-about introduction, take a look at the table of contents, choose the topics that interest you the most and find out what both the developers and the authorities have to say about it. You won't be disappointed.

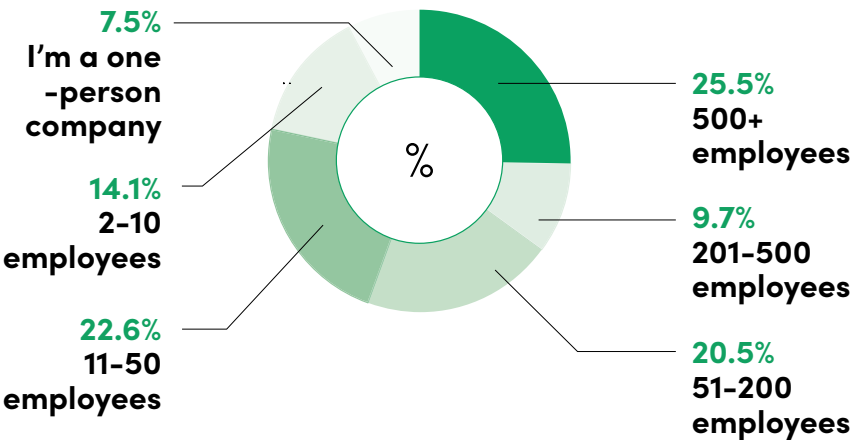
## For how long have you been in the frontend development game?



# How would you describe your seniority?



# How big is the company you are working in?



02.

# Frameworks

React is king.

But who's the contender?



**Dylan Schiemann**

CEO of [Living Spec](#), Co-creator of [Dojo](#)



## **One thing is certain: React dominates the mind share today.**

When you look at the results of the State of Frontend survey, one thing is certain: React dominates JavaScript framework mind share today. However, at the same time, it seems that next-generation reactive frameworks may soon rise as lean alternatives to the React ecosystem. And it all has much to do with the rising popularity of TypeScript.

For the past several years JavaScript developers have gravitated towards React, Vue.js and Angular as the leading frameworks. Relative interest in Angular has decreased due in part to the long delay in shipping Ivy and, similarly, interest in Vue has stalled a bit due to the long-awaited and somewhat delayed Vue 3.0 release. It all helped React dominate the JavaScript framework market with 74.2% of the survey's respondents using it – more than Angular and Vue.js users combined!

It doesn't mean, however, that the React community lives without a care in the world. A major change took place recently when developers started turning away from Redux. We can already see that, when it comes to state management, more people use React Context API and hooks (49.6%) than Redux (48.2%) – of course, some of them still use both but the trend is visible. Also, as a side note, while discussing big JavaScript frameworks is important, we must not forget about jQuery which, while rarely talked about, still remains the most widely deployed JavaScript library on the web.

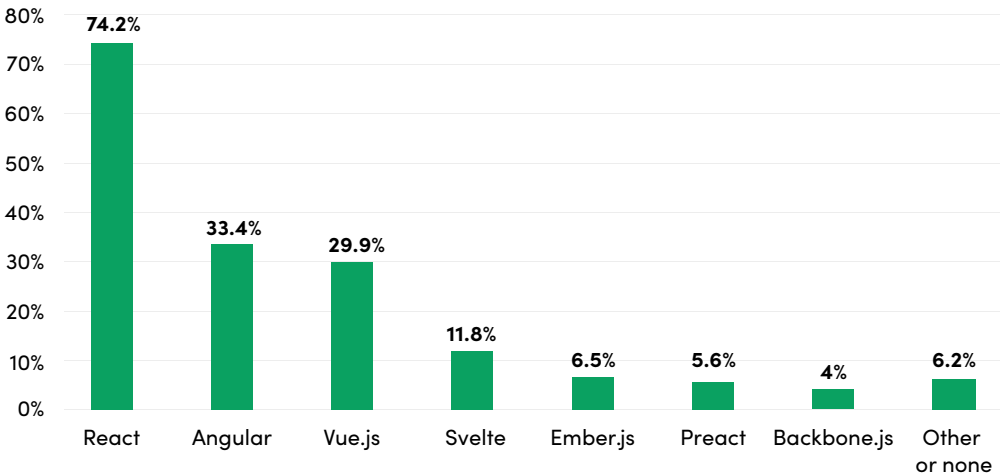
And what about the future of JavaScript? We're already seeing significant interest in next-generation reactive frameworks such as Svelte which strives to provide reactivity on top of normal DOM structures. Yet another competitor is Stencil – a framework focused on web components and, just like Svelte, on efficient compilation. Also, Dojo has re-emerged as a reactive TypeScript-first framework promising intelligent defaults for faster out-of-the-box experience.

Some argue that these next-generation frameworks may be great for smaller applications but require more work when building large apps. It's true that all of them provide much smaller default application bundle sizes as they do not carry the same legacy as frameworks which need to support features of the past few years. Also, they are very aligned with modern standards and language features.

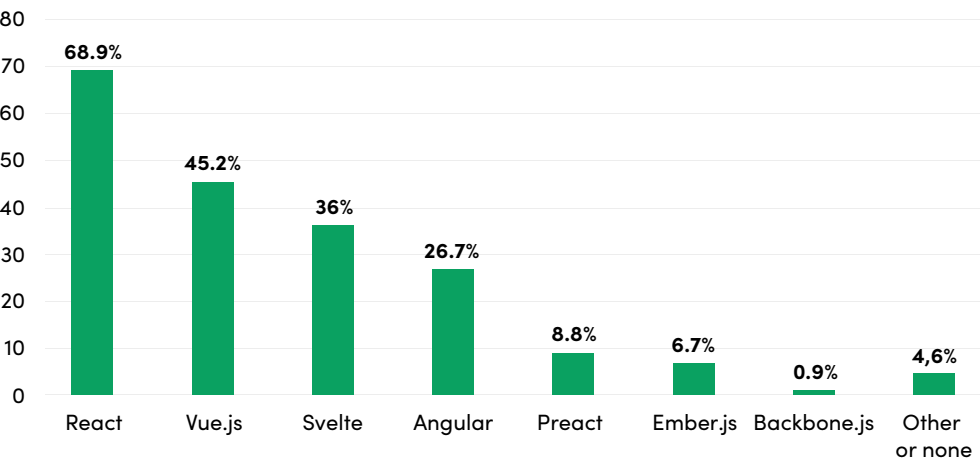
And this is where we must discuss the growing importance of TypeScript. With 77.2% of respondents already using TypeScript and most of them preferring it to JavaScript, it's not surprising that frameworks are improving their support for TypeScript and many start to leverage TypeScript internally. It's true for both the already established frameworks (like React and Angular) and the next-generation ones (Stencil and Dojo in particular).

With all these changes going on, I'm really looking forward to seeing what happens next in the realm of JavaScript frameworks. Because one thing is for sure: React is the king now but there are already a few contenders for the throne waiting.

# Which of these frameworks have you used during the last year?

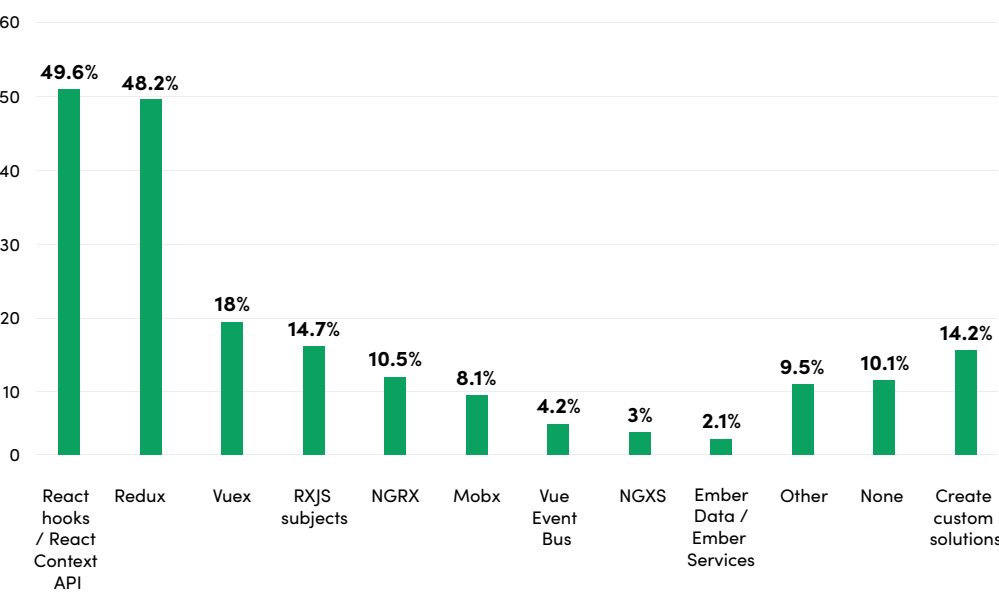


# Which of these frameworks would you like to keep on using or want to learn in the future?

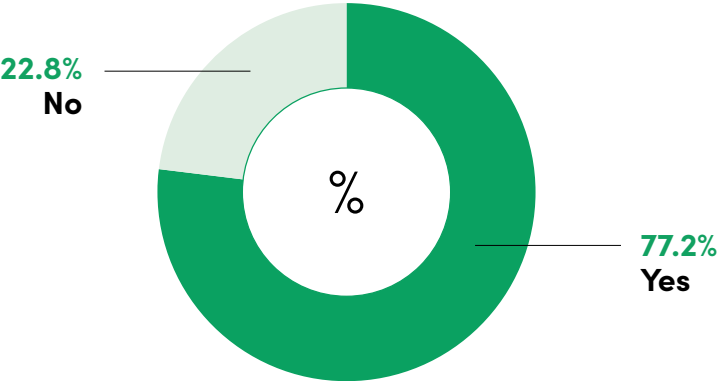




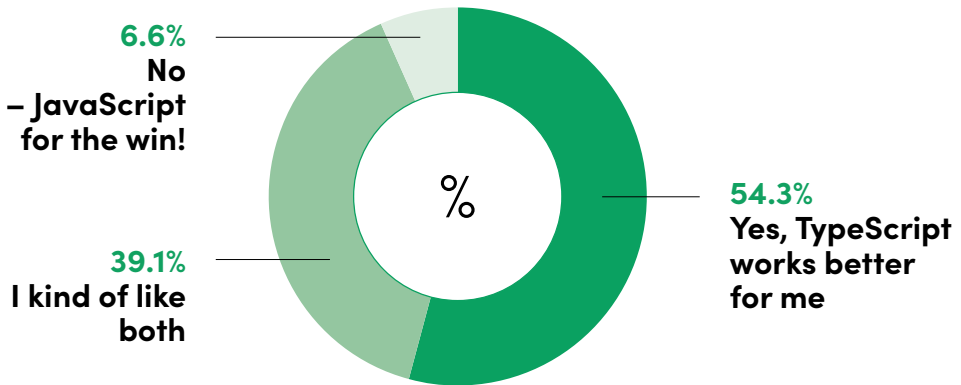
# Which solutions do you use when it comes to state management?



# Have you used TypeScript during the last year?



## Do you like TypeScript better than JavaScript?



## What do you think about the future of TypeScript?

