Need to volunteer? (Interested in hosting, researching, editing or be part of my team please reach out to me)

(https://www.mappingthejourney.com/contact/) Subscribe on iTunes (https://itunes.apple.com/us/podcast/mapping-the-journey/id1246370704?mt=2) Subscribe on Soundcloud (https://soundcloud.com/pramod-shashidhara) Listen on Google Play Music (https://play.google.com/music/m/Iwjeczy53fj5brw4nbbwatijzcy?t=Mapping_The_Journey) AVAILABLE ON STITCHER (http://www.stitcher.com/s?fid=141980&refid=stpr) (https://play.pocketcasts.com/) **Donate for Inspiration**

Episode 8: Interview with Ryan Dahl, Creator of Node.js

()

August 31, 2017 | Pramod HS ()

DOWNLOAD (http://traffic.libsyn.com/mappingthejourney/Episode8.mp3)

(/#facebook) (/#google_plus) (/#reddit) (/#twitter) (/#linkedin) (/#whatsapp) (/#email) 赞 617

Post Views: 81,862

Ryan Dahl is a Software Engineer working at Google Brain. He is the creator of Node.js, JavaScript runtime built on Chrome’s V8 JavaScript engine. Currently, he is working on deep learBrowserifyning research projects. His focus is mostly on image-to-image transformations like colorization and super-resolution. He has contributed to several open source projects including HTTP Parser, libuv.

Pramod: Hello People. Welcome to mapping the journey. When we hear about Node.js, it’s Ryan Dahl. He showed us that we are doing I/O completely wrong and also taught us how to build software using pure async programming model. Today’s guest is the man himself Ryan Dahl, hacker, brilliant programmer and creator of Node. I am very excited and honored to have you on the show. Welcome Ryan

Ryan: Hello! Nice to be there… here.

Pramod: Ryan we know you as Creator of Node, tell us about your life before Tech?

Ryan: Sure. I grew up in San Diego; my mom had got an Apple 2C when I was six years old, so I guess I’ve had kind of early access to computers. I’m 36, by the way. So, I came of age just as the internet was coming out. I went to community college in San Diego and then went to UCSD afterward, where I studied math. Then, I went to grad school for Math, at the University of Rochester.

Yeah. There, I studied algebraic topology, which was a very abstract subject, that I found very beautiful for a couple of years, but later I got bored of it because it was not so applicable to real life, it seemed. After grad school, well… so, that was a Ph.D. program. And once I realized that I did not want to be a mathematician for the rest of my life, I dropped out of that program. And bought a one-way ticket to South America and went there for a year, where I was kind of in starving student mode, and found a job doing some websites with this guy, Eric. And that’s kind of how my programming career started. It was working on the Ruby on Rails website for a snowboard company.

Pramod*:* Nice! That must be an experience right to drop out of the Ph.D. program, travel to South America and working as a Web Dev.

Ryan: Yeah. I mean, so… coming from grad school, you're used to dealing with very abstract problems, and working on websites was a very concrete process. But I was trying to make it into a beautiful mathematical theory like I was exposed to at grad school. And I think that got me thinking about… I guess I liked how Ruby made development so much more, I guess, you could express your ideas more clearly in Ruby. And that was interesting at the time. And I think Rails was impressive in that. It gave this new structure, and probably, it wasn't new, but I think Rails popularized the structure of model view controller. And you know, those two things combined, really was kind of interesting to me.

Pramod*: Yes, building web applications, is very interesting. And Ruby is a perfect tool. Next, you continued working as a freelance web developer in Germany. One of the projects you worked on was Node. And I think you continued working on it for the next six to eight months?*

Ryan: Right. So, after South America, I moved with my girlfriend to Germany, because she was German and had to return to university. And I started going to Ruby conferences there, where people were talking about this new paradigm of model view controller. And one of the guys there was Chris Neukirchen if I'm pronouncing that correctly. And he developed this project called Rack, which was a simplified abstraction of a web server. So, it turned a web server into a single-function interface, where you get a request, and then you return a response.

That, combined with some freelance work that I was doing on Nginx Module, for Engineyard, got me thinking about how… let me step back a second. In Nginx, everything is asynchronous. So, when you build a module for it, you have to be very careful to be non-blocking. And yeah, I think the combination of Chris Neukirchen's rack plus how Nginx structured its web server with non-blocking IO, led me to start thinking about how those two things could be combined.

Pramod*: Now you sort of had the idea with Rack and Nginx. "How did you convince yourself, okay I will spend my next 6 months building framework that can run Javascript on the server side, which may increase the performance a great deal?*

Ryan: So, those two pieces that kind of simplified web server interface, which was Rack, and the asynchronous part, which was Nginx, I had been thinking about. And then Chrome was released in December 2008. And along with that was the V8 Javascript interpreter. I shouldn't say, interpreter. It's a jitted run-time. So, when V8 came out, I started poking around with it, and it looked fascinating and clean, and fast, and suddenly, I clicked that: oh! Javascript is single-threaded, and everybody is already doing non-blocking.

I'm using my fingers to do air quotes, but like in the web browser, people are already making non-blocking requests when they make AJAX request and stuff. And I thought: oh, wow! I think JavaScript plus asynchronous IO plus some HTTP server stuff would be a cool thing. And I was so excited about that idea that I just worked on it non-stop for the next four years.

Pramod*: Yes Javascript plus async i/o worked really well. I believe developers were waiting to see a framework that did that. And Just curious, during this time, was there any mentor or did you ever consult with anyone"? or Was it just you?*

Ryan: It was just me. I had some programming friends who gave advice, and definitely… I mean, the first bit of it was just me in my room. But later on, I ended up moving to San Francisco and working at Joyent and meeting lots of really great programming professionals. And yeah, many people mentored and gave ideas that contributed to Node after that.

Pramod: Nice. Take us through the journey you went through with the development of Node. I know it's a long time Ryan since you created Node around 2009.

Ryan: I think at least for myself, there's no greater moment in my life than when I'm like, in the flow, and having an idea that I believe in. And have the time to sit down and work on it. And I think Node was an idea waiting to happen and had I not done it; somebody else would have. But it just so happened that I was relatively unemployed and had some free time, and could work on it non-stop for months, which is what was required to get an initial product out. So yeah, that was awesome, it was fun.

Pramod*:* Great. That's fantastic. You did it really well. Node is built on the idea of "pure async" programming model. How did this idea work out for node?

Ryan: Yeah, I think that's a really interesting question. Now, it's been several years, and I haven't worked on Node myself since like, 2012, or 2013. And Node, of course, is a big project at this point. So, yeah, I think… when it first came out, I went around and gave a bunch of talks, trying to convince people that they should. That maybe we were doing I/O wrong and that maybe if we did everything in a non-blocking way, that we would solve a lot of the difficulties with programming. Like, perhaps we could forget about threads entirely and only use process abstractions and serialized communications. But within a single process, we could handle many, many requests by being completely asynchronous. I believe strongly in this idea at the time, but over the past couple of years, I think that's probably not the end-all and be-all idea for programming. In particular, when Go came out.

Well, I think Go came out a long time ago, but when I first started hearing about Go, which was around 2012. They had a very nice runtime that had proper green threads and easy to use abstractions around that. That I think makes blocking I/O – again, blocking I/O in quotes, because it's all in green threads at the interface of… between Go and the operating system, I think it is all non-blocking I/O.

But the interface that they present to the user is blocking, and I think that that's a nicer programming model. And you can think through what you're doing in many situations more easily if it's blocking. You know, if you have a bunch of following actions, it's nice to be able to say: do thing A, wait for a response, maybe error out. Do thing B, wait for a response, error out. And in Node, that's harder, because you have to jump into another function call.

Pramod*:* Yeah, I like the programming model of Go. Using goroutines is so much easy and fun. In fact, we are using at work for building a distributed application.

Ryan: Yeah, I think it's… for a particular class of application, which is like, if you're building a server, I can't imagine using anything other than Go. That said, I think Node's non-blocking paradigm worked out well for JavaScript, where you don't have threads. And I think that a lot of the problems with kind of the call-back soup problem, where you have to jump into many anonymous functions to complete what you're doing has been alleviated these days, with the async keyword, the async feature that's in Javascript now.

So, kind of the newer versions of Javascript has made this easier. That said, I think Node is not the best system to build a massive server web. I would use Go for that. And honestly, that's the reason why I left Node. It was the realization that: oh, actually, this is not the best server-side system ever.

Yeah. I think where Node has shined, weirdly, on the client side. So, doing kind of scripting around building websites. So, Browserify, for example. Kind of bundles up client-side Javascript. So, you can have all this server-side processing of client-side Javascript. And then, you know, maybe small servers to… maybe little development servers, and here and there, maybe some real servers serving live traffic. Node can be useful, or it can be the right choice for it. But if you're building a massively distributed DNS server, I would not choose Node.

Pramod: This should be a good takeaway for all the developers around the world. Choosing a right tool for application is so much important. You are not biased at all with Node. You introduced Node.js to the world in JsConf 2009 Berlin. Were you surprised with the success and traction it suddenly received?

Ryan: Yeah. I mean, I was basically in a continual state of a surprise for four years. Because it grew very fast, and people liked it very much. So yeah.

Pramod: Thereafter you joined Joyant and worked on Node full time and you moved to SF right? How was the experience? Developers loved it and You were the center of it all.

Ryan: Definitely, it was an experience of a lifetime, and I felt in the center of it all, being at conferences and whatnot. At one point, I went to Japan, and people were asking to take their photo with me, and I realized… I don't know; I felt kind of weird about that. Also online, I think around that time, I felt like whenever I commented on something, I would get like, 100 responses from people.

And so, I found that I had to choose my words very carefully and how I presented myself because it seems like people were listening, which was strange. And I didn't like that aspect of it. I mean, I'm a programmer, and I want to write code and sometimes share my opinion without thinking too carefully about it. And so, I think I'm not one to… yeah. I didn't enjoy that aspect of it so much.

Pramod: You were what, 29, 30, when you introduced Node? And Node made such a massive impact.

Ryan: Yeah. I mean, I was definitely a more of a novice developer at that point.

Pramod: Ok. Ryan, there were many server-side javascript projects at the same time. Node was not the only one. What do you attribute the success of Node?

Ryan: Right. So, there were several people, that were trying to get the server side JavaScript thing going. I can't even enumerate them anymore, and I forgot what they are. Well, whatever.

The thing is that they were all doing blocking I/O, and that didn't jive with how Javascript is structured because you don't have threads. And so, if you're doing blocking I/O, you literally can't serve requests. Like, you're doing one at a time, and that's just never going to work. That, plus the fact that I like, sat down and made the HTTP server work well. So, I had a demo where you could… I had an HTTP server, and then a raw TCP server. And I made those things work well so that people could sit down and build a website without going through too much hassle.

And honestly, building a web server is not the easiest thing, and I think a lot of these systems kind of left to their community to make that, and thus, nobody did. Because there is nothing to use the system with. I think it's important that when you release a software framework, or maybe any software, that you have a demo that people can sit down and use immediately. And that was one of the things that went right with Node; it was that people could download it and use the web server directly.

Pramod*:* Yes. Good Demos & if people could download, install and use it easily makes a big difference. Also, people knew javascript, they could start coding in no time. When I started working on node, it was that much more easy as I knew javascript well.

Ryan: Yeah. I think we take for granted how easy it is to switch between languages. I mean, even if you know another language, making that context, which is pretty difficult. And there's a lot of people who are very familiar with Javascript. And giving them these tools to be able to use it in other contexts excites people. You suddenly can do it a lot more than you were able to do before.

Pramod*:* Yes. In 2012 node already had a huge developer base. Why did you step away from, turning over the reins to Joyent's Isaac Schlueter?

Ryan: Yeah. Right. So, I mean, I think it was a combination of a couple of things. So, I think that the main thing was that, at that point, I had been working on Node for four years. And I had gotten to the point where I wanted it. I never wanted Node to be a massive API. I wanted it to be this kind of small, compact, core that people could build modules on top of.

And there was a couple of the main things that… key features that I wanted to support. So, extension modules were added early on; we got all the networking libraries worked out, HTTP, UDP, TCP, we could access all the file systems.

And then, kind of a big chunk, which was maybe a year of work with like, five people, was putting it to Windows and doing that properly. And that we wanted to use Windows abstractions for asynchronous IO, which is their IO completion ports. And so, that required rewriting the core library, which, the result of that was the libuv library.

Yeah, but at some point, all of that was done, and we had released on Windows. And you know, it was kind of at the point where it's like: ok. I mean, this is what I had intended to create, and I'm pleased I got the chance to follow through with it. And of course, there's going to be, you know, an infinite amount of bugs to fix for the rest of my life, but… you see, there are enough people involved to that. I don't need to do that necessarily, and I would like to do other things. So, that plus the fact that Go came out, and I didn't see Node as being the ultimate solution to servers. And then, yeah, also just not wanting to be at the center of attention whenever I made a blog post.

Pramod*:* Nice. Yes, Some people do not enjoy being in the limelight. When you started working on Node, you definitely had some goals. How is Node.js today stacked up against it?

Ryan*:* I mean… Node is used by hundreds of thousands, if not millions of people at this point, and I think it's certainly exceeded any expectation of what I thought would happen with it. Yeah, it's cool.

Pramod*:* Ryan after your wonderful journey with Node, what did you decide to work on?

Ryan: So, after Node, I moved… after I left Joyent and quit the Node project, I moved to New York, and took

Ryan: So, after Node, I moved… after I left Joyent and quit the Node project, I moved to New York, and took some time off to work on my projects. So, I had a bunch of projects. You know, at the time, Instagram had come out, and it was new, and it seemed simple, and everybody was saying: wow, that's so simple, I could have built that. And I couldn't help but think the same thing. So, I had a social network project; I had a build system project for C++, I had another build system project for HTML, which was kind of like, Browserify, which would kind of package up your Javascript and HTML, but in a smarter way.

Yeah, I had a bunch of projects, none of which panned out, really, in my mind. Although I think some of them are currently still on the backburner, like my social network project, which I will get back to at some point. Yeah, I was doing that for a while. And then I started reading about… well, I began hearing about convolutional networks and how image classification had been solved, and that got me interested in machine learning.

Pramod*: Also you were part of Google Brain's Residency program. How was that experience?*

Ryan: Yeah. So, I just spent a year out in Mountain View. So, stepping back a second. So, TensorFlow released two years ago now.

And with it, they announced this Google Brain residency program, where they invite like, 20 people to come to Google Brain, which is one of Google's machine learning research labs. And people… I think the idea with it was not necessarily people who had studied machine learning but had some background in Math and programming and were interested in machine learning to like, come and kind of play around with these new ideas. Because machine learning is changing fast, and there is a large body of work that has been done.

But now that the community has narrowed in on neural networks as being the most useful algorithm for machine learning, that maybe just bringing in a bunch of people and just playing around with that, and this new ML framework, called TensorFlow, would result in some interesting ideas. So yeah, I spent a year there, basically programming models and writing papers about those models. I worked on the mostly image to image transformation problems. So, you know, if you have some input image and you want to predict some output image. I find this problem interesting, because, for example… let me give some examples.

So, the problem of colorization. You can take a black and white photo as input, and you can try to predict the colors of the picture as an output. What's cool about this problem is that there are unlimited training data. You can take any color photo and de-saturate it, and then that's your input image, right?

So, one of the problems of machine learning is that you need lots of data, and with these sort of tasks, that problem is not an issue. And also, there's been a lot of work in generative models recently, that is models that output images. In particular, there have been generative adversarial networks, and pixel CNN, which has demonstrated the ability to learn the manifold of natural images, that is like, really understand what a real image is and what is not an actual image, what looks like a real image.

So yeah, my idea was to kind of take this recent work in generative models and take this unlimited training data and see if we can do some image transformation problems. So, I did some work on super-resolution, which is taking a low-resolution image and increasing the resolution. That's also an image to image transformation problem. And I've done two projects now on colorization.

Pramod: Nice explanation Ryan. Yes, I have read that tensor-flow has been a great platform for many machine learning problems. Image classification, transformation. I don't get it much, but I believe it's

machine learning problems. Image classification, transformation, I don't get it much, but I believe it's fascinating. Are you continuing your work with ML?

Ryan: Right. So now, I'm still at Google, as a software engineer, working on the same sort of problems. Studying generative models and trying to help the researchers build the next generation systems, next-generation models.

Pramod: Nice Generative models that are so much different than javascript, node or web-development work you did before.

Ryan*:* Yeah, I guess so. But I also started in Math, so I have a fairly decent foundation of math knowledge, I guess. And yeah, I guess… I think people like to kind of bend other people into certain areas, and I don't really wanna do that. I don't wanna be a Javascript person, and I don't wanna be a machine learning person. You know, I think people… it's interesting to just explore what's possible. What's exciting is building something new that hasn't been done before that could benefit humanity in some way.

Pramod*:* Nice. Yes, nice to know that Machine learning requires a good math background. In one of your recent blogs on Optimistic Nihilism, you say that we would be able to someday emulate brains and build a machine that understands and thinks like humans do. How far are we in achieving that?

Ryan: Yeah. I have to be a bit careful about kind of prophesying… I mean, this is my opinion. We are nowhere near matching human intelligence. I mean, the machine learning systems that we're using are very, very simplistic, and don't work at all. In fact, I have a blog post about my residency, in which I enumerate all the difficulties there are in developing these models. I think people who don't work in the field have this idea that you can kind of take this model and push some data through it, and it's just going to work. But that's not the case. These things are very finicky and not well understood, and it takes many, many months of careful tweaking and experiments to get even the most modest results.

So, we are nowhere near it, but that said, I think the foundation… there's some promising technology that has been improved recently, which is namely that convolutional networks seem to work, and that propagation appears to work. And the fact that these things are based on a model, this neural network model, that is not brain-like, but the brain somehow inspires it, is pretty enticing. We also have GPUs, and we showed how we could train on these and distribute training across GPUs to some extent. So, I think the kind of… the foundations of building bigger, smarter systems is happening. And personally, I'm an atheist, and I believe that there's nothing more in my brain other than the chemicals, and neurons that are the matter of my brain. And I think that my consciousness, all of our consciousnesses are encoded, somehow, in interactions between those neurons. So, I don't see why we wouldn't be able to, someday, with enough research and work in this field, emulate that sort of behavior. It's too far out to predict how long that would be.

Pramod*:* Great. You have seen it all Ryan and Where do you want to see tech in next 20 years?

Ryan: I mean, I am very excited about machine learning and the possibilities that it brings. I think that even before, like, way before we get to real artificial intelligence, that there are many applications of this technology. I mean, any system where you could… wear a smart glass would help you, is going to benefit enormously from this technology. So, there are just uncountable industrial processes that could take advantage of this sort of thing, you know. I think recycling centers, with sorting… sorting recycling with computer vision, and what not. I mean, there's just many, many systems that could benefit from simple machine learning systems. And I think that we're going to, more and more, see these systems get applied to different processes. So, I think that's going to affect the technology sector significantly, and all of the humanity greatly.

Pramod*:* Yes machine learning is very exciting. I get so much excited when I see autonomous cars in Mountain View. Someday I would like to sit back and give complete control. Thank you, Ryan, for providing us with this nice framework Node and thanks for being on the show. Also good luck with your future projects. It was wonderful speaking to you

Ryan***:*** Yeah, great. Thanks for having me, it's fun to talk about it.

Pramod*:* Thank you. That is it, listeners. I really enjoyed speaking to Ryan, such a humble and awesome guy. He has achieved so much in his early years in tech. Such an inspiring story. Bye for now, I will meet you all in 2 weeks time with another interesting journey. Shukriya.

(/#facebook) (/#google_plus) (/#reddit) (/#twitter) (/#linkedin) (/#whatsapp) (/#email) 赞 617

Leave a Reply

11 Comments on "Episode 8: Interview with Ryan Dahl, Creator of Node.js"

Notify of new follow-up comments Email ›

Join the discussion

Sort by: newest|oldest|most voted

francoisromain

Guest

Very interesting to read, thank you. I think Ryan talks about Browserify (not Browser FI).

7 REPLY

11 months 27 days ago

Pramod Shashidhara (https://mappingthejourney.com/single-post/author/pramodbs/)

(https://mappingthejourney.com/single-

Thank You. Fixed 🙂