Using Node.js to Deliver NPR via the Internet of Things

> Tasked with building products for the fast-growing voice assistant platform Amazon Alexa, developers at NPR turned to Node.js. Drawn by a robust Node.js SDK for Alexa, as well as the technology's compatibility with serverless infrastructure, the team developed its first Alexa app in two and a half months, notching both cost and time savings. Another win: Node.js will enable NPR to reuse code as it expands to other voice assistant platforms.

For the 15 million listeners who start their morning with National Public Radio, tuning in is now as easy as saying, "Alexa, play NPR."

With nearly half of Americans using voice assistant platforms, and sales of Amazon Echo and Google Home devices reaching into the tens of millions per year, NPR set out in the fall of 2017 to build on its foothold in the Internet of Things. The company tasked a small, agile team—two developers, one designer, a product owner, and a scrum master—to create products for voice assistant platforms, starting with that one Alexa app, known as a "skill."

Historically, NPR had two fairly large monolithic code bases, primarily PHP and some Java. But in the past few years, an influx of new engineers to the company brought with them experience with different languages and tools. "There's been a definite shift to being more open to other technologies," says Senior Web Developer Nara Kasbergen, who leads the voice assistant platform team.

Finding the Right Technology

For this greenfield Alexa development, Kasbergen had a sense early on that she wanted to use JavaScript and Node.js, and she had the support of her manager. “Our teams are fairly autonomous in our technology choices,” says Software Development Manager Olubunmi Odumade. “Our developers enjoy working with Node.js, and if they’re happy, we managers are happy.”

For several reasons, Node.js seemed like a good fit for the project. While the Alexa ecosystem has multiple SDKs, “the Node.js one is the most fleshed out,” says Kasbergen. Plus, says her teammate, Web Developer Tommy O’Keefe, “One of the things that always pushes me in the JavaScript and Node.js direction is how easy it is to use npm. Then there’s the ease of information if you run into blockers. It’s just such an active community.”

> While the Alexa ecosystem has multiple SDKs, “the Node.js one is the most fleshed out,” says Kasbergen. Plus, says her teammate, Web Developer Tommy O’Keefe, “One of the things that always pushes me in the JavaScript and Node.js direction is how easy it is to use npm. Then there’s the ease of information if you run into blockers. It’s just such an active community.”

Still, the team evaluated other possibilities, including using PHP. “But our conclusion was, ‘Why would you use PHP for this when it seems like the Node.js tool is really good?’” says Kasbergen.

The other big factor was that Alexa and the other voice assistant platforms all recommend that developers use a serverless infrastructure when building skills. NPR chose the AWS Lambda serverless computing platform, and at the time, Nasbergen says, “the options were to use JavaScript, Python, or Java. Python is the one thing that we don’t really do here at NPR, and Java is something that we do, but no one really loves it. It was a fairly easy decision to stick with JavaScript and Node.js.”

Building Skills and Templates

Though they had both dabbled with the Alexa platform, neither had built a skill before. So for the first few weeks of the project, Kasbergen and O'Keefe worked on building a prototype that streams music.

"We just wanted an opportunity to try out the Alexa SDK and some of their build tools and see how that works," says Kasbergen. "One of the biggest hurdles was that we had to learn to interface with the Alexa audio player. The documentation was fairly verbose and not necessarily the best way to wrap your head around what you need to do to get Alexa to play audio. So we practiced with the music skill."

With that under their belt, they felt they had a good idea of how they wanted to structure the code base of the skill that would actually go into production. "We knew that although we were only focused on Alexa, there was a decent chance that at some point, we would be asked to build the same thing for Google Home," says Kasbergen. "So we used one code base and created separate folders: one for logic that would potentially be shared between Alexa and Google Home, and the other for all the code that was specific to Alexa."

Node.js' Portability Across Voice Assistant Platforms

The team built the "Play NPR" skill in two and a half months, and followed up with the skill for NPR One (the company's cross-platform app for radio news, stories, and podcasts). Both went live in the spring of 2018.

They will soon test their hypotheses about how code can be shared across platforms; their next project is developing the NPR One skill for Google Assistant. "We're hoping to prove that we can write as little code as possible to get this working on Google Assistant," says Kasbergen. "Secondarily, we want to think about how we can apply those learnings in the future, developing either an actual framework or at least a framework in our heads. We use Jest, our testing framework of choice, and we decided to bring in TypeScript. We're setting up templates for ourselves to make it easier to do this work in the future."

With that goal, "We're fairly certain that the choice of using Node.js was a good one," says Kasbergen. "All of these voice assistant platforms really are leaning on serverless technologies, and the only technology or language they had in common was Node.js."

> With that goal, "We're fairly certain that the choice of using Node.js was a good one," says Kasbergen. "All of these voice assistant platforms really are leaning on serverless technologies, and the only technology or language they had in common was Node.js."

O'Keefe thinks the reason is simple. "Certain environments are going to be more conducive to quickly getting started when they need to fire up a new container, and Node.js seems to be a good choice in that space," he says. "And with the serverless code, you have to write things in a stateless way, and I think JavaScript is conducive to that. It's also a language that's popular and spans both front and backend and serverless. There's a lot that you don't have to understand."

Impact and Savings

Choosing Node.js for NPR's foray into voice assistant platforms had a positive impact on the bottom line. "Using Node.js allowed us to use AWS Lambda," says Odumade, the NPR Software Development Manager, "and going serverless has led to lower costs than running a traditional web server." Kasbergen estimates that the savings add up to at least an order of magnitude.

NPR's traffic peaks in the morning, as listeners wake up and commute to work, and then sees a smaller bump around the time people go home in the evening. "The fact that we can essentially spin down our capacities overnight and then spin them up again in the morning to account for that morning traffic spike is a huge cost saving for us," says Kasbergen. "If we ran a traditional web server, we'd have to scale for that peak, which would leave us with unused capacity overnight."

Additionally, going serverless meant "we haven't really had to involve our system admins in this project," she adds. "Right now, sysadmin time is one of the most scarce resources in the department. It's a huge benefit to the entire organization, and why pretty much everyone else is onboard with us taking this approach."

For O'Keefe and Kasbergen, using Node.js also enabled them to do their work faster. "We're comfortable with the Node.js and npm ecosystem," says Kasbergen. "Overall, it just seems like our code is fairly minimal. We don't spend a lot of time dealing with the framework headaches."

> For O'Keefe and Kasbergen, using Node.js also enabled them to do their work faster. "We're comfortable with the Node.js and npm ecosystem," says Kasbergen. "Overall, it just seems like our code is fairly minimal. We don't spend a lot of time dealing with the framework headaches."

Best Practices

Almost a year into their work developing apps for voice assistant platforms using Node.js, the two are happy to share their experiences. Both came from a frontend background, and Kasbergen says being able to use JavaScript on both the front and backend with Node.js makes her "feel like I still get to flex all my JavaScript muscles that I've developed over time, which is really satisfying." O'Keefe adds: "JavaScript feels fluent for me at this point. Beyond that, when I think of where the industry as a whole is, I feel like I'm part of the current cutting edge, which is great and fun to be involved in."

Kasbergen's advice for anyone starting with Node.js is to go straight to ES6 syntax: "There's a lot of great resources for it, so you don't have to worry too much about ES5. If you're really looking to level up your development ability, I would definitely encourage people to look into TypeScript. That ecosystem and the tooling have improved so much over the past year, it's just so much easier to get that set up in a Node.js project than it used to be."

For O'Keefe, using a version manager like nvm is key. "Make sure that the version of Node.js that you're running locally is the same version that you're going to be using in production, and lock down your dependencies to a specific version," he says. Aside from that, he says managing and deploying Node.js apps is "pretty low-friction, easy, and pain-free."

> For O'Keefe, using a version manager like nvm is key. "Make sure that the version of Node.js that you're running locally is the same version that you're going to be using in production, and lock down your dependencies to a specific version," he says. Aside from that, he says managing and deploying Node.js apps is "pretty low-friction, easy, and pain-free."

Planning for the Future

At this point, the NPR One Alexa skill code base has 100% test coverage, and that's something O'Keefe and Kasbergen are both proud to have accomplished so smoothly. Especially since NPR considers this IoT technology business-critical.

"The skills we built have allowed us to extend our reach and further engage with people who are already listeners," says O'Keefe. "For us, ideally, we're seeing a voice platform device as a radio replacement, and a new way that listeners can interact with our audio content. It's a platform that's going to keep changing, but NPR is definitely committed to being in that space."

> "The skills we built have allowed us to extend our reach and further engage with people who are already listeners," says O'Keefe. "For us, ideally, we're seeing a voice platform device as a radio replacement, and a new way that listeners can interact with our audio content. It's a platform that's going to keep changing, but NPR is definitely committed to being in that space."

And with the foundation they've established with Node.js, they're confident they'll be prepared for any other voice assistant platform that emerges. "Samsung has Bixby, Microsoft has Cortana, and Apple has Siri," O'Keefe points out. "All of those developer ecosystems are still fledgling compared to Alexa and Google Home. We don't necessarily have a project for them on the roadmap yet, but we want to be ready if they come to us tomorrow and say, 'Hey, our developer platform is ready.'"