Ben Forshey

Advanced Server-Side Languages 1506

Information about the framework.

I'm choosing Node.js, which is a server-side JavaScript platform built on Chrome's V8 engine. As described on Node.js' homepage, "Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices."

Why you have selected this framework?

I selected this framework for the following two reasons:

1) I want to be very proficient at JavaScript, because I've found it to be extremely useful with client-side development. If I can learn JavaScript better by using it for client-side and server-side development, I'll be a better developer. There's a mental cost to switching languages mid-stream. Even as an inexperienced developer, I can see that. A few days ago, while programming in PHP for the back-end of a side project, I couldn't remember the slight difference between a PHP associative array (since PHP 5.3) and a JavaScript multidimensional array. This problem was quickly solved with a quick syntax lookup, but the better scenario

would be if I could not need to remember that in my day-to-day work. My (limited) mental resources could be focused on getting a lot more proficient with JavaScript.

2) Node.js get's a lot of rave reviews over PHP from a lot of people for its non-blocking event driven model (in the same way that NGINX gets a lot of attention over Apache2). I know next to nothing about Node.js, but I think that now is the time to jump in.

What are you planning to develop and how this framework will help you achieve this goal.

I'm planning on developing a web-based catechism app, inspired from http://www.newcitycatechism.com/home.php. It bugs me that such a great tool gets second-class treatment on the Web. I'd like to try and change that. Also, there's a native app for the iPad (designed by a professional team), but the ports of that for the iPhone OS and Android OS are pretty rough. I'd like to represent the Web well, by making something that's available on any HTTP-enabled device.

List websites containing tutorials, documentation, and blogs of this framework.

I probably won't need all of these, but here's a list that I've compiled so far:

Vagrant (maybe for my dev environment)

https://docs.vagrantup.com/v2/getting-started/index.html

http://thejackalofjavascript.com/vagrant-mean-box/

Node JS

http://webapplog.com/php-vs-node-js/

https://strongloop.com/strongblog/node-js-php-get-started/

https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-an-ubuntu-14-04-server

http://hectorcorrea.com/#/blog/introduction-to-node-js/51

Web Frameworks / Middleware / CMS

http://keystonejs.com

http://expressjs.com

http://hapijs.com

http://passportjs.org

http://mongoosejs.com/index.html

Databases

https://www.mongodb.org

http://code.tutsplus.com/tutorials/getting-started-with-mongodb-part-1--net-22879

Authentication / Login

https://scotch.io/courses/easy-node-authentication

http://code.tutsplus.com/tutorials/authenticating-nodejs-applications-with-passport--cms-21619

http://code.tutsplus.com/articles/social-authentication-for-nodejs-apps-with-passport--cms-21618

Templating

http://jade-lang.com

http://olado.github.io/doT/

List websites built by this framework.

The following websites use Node.js (list complied from CoderFactory and nodejs.org):

- Paypal
- eBay
- Microsoft
- LinkedIn
- Yahoo
- Groupon

- Wall Street Journal Online
- Klout
- Secret
- BlandPage
- Geeklist
- Opencare
- Shutterstock
- Storify
- Ancestry
- Quizlet
- Zendesk
- LearnBoost