

What do you do to stay productive? Any tool or processes you can recommend?

Azat: I use Reminders in iOS. I have one list that is “urgent” and another that is not urgent. I also use Evernote (evernote.com) and Notes in iOS and Mac OS X. I use Google spreadsheets because I like to keep things simple.

Can you recommend any advice for freelancing with Node.js? Are there any useful websites?

Azat: Yes. Airpair (airpair.com) is a website for experts. You can join as a developer or to look for a developer. You could also use oDesk (odesk.com) or Freelancer (freelancer.com).

I would like to use a VPS to upload my node apps.

Azat: There are multiple solutions. You could use Joyent or Amazon, as long as you have the UNIX box or similar. Installing Node.js is just one command line. I have a chapter in my previous book Practical Node.js devoted to deployment. The deployment to (Amazon Web Services) AWS took less time to install, but it is not as secure or as stable as Heroku (PaaS).

How do you setup a server to run Node apps? Do you use forever or have any recommendations about running a blog on a VPS like DigitalOcean?

Azat: For Digital Ocean you can use a droplet like Docker. Use Forever (an NPM module that installs globally) and Upstart (script that lives in your UNIX system). Forever will restart Node.js apps when they crash. Upstart will restart the Forever command. Upstart will restart the Forever itself, if the whole box is down or if you redeploy. It's good to use both. In some older applications, where Upstart is not available, you can use /etc/init.d.

So Practical node.js has information on deployment?

Azat: Yes. Practical Node.js has a chapter on deployment, both platform as a service and infrastructure as a service. Platform as a service is Heroku and infrastructure as a service is AWS. Rapid prototyping with JS also has platform as a service. You could also use Varnish cache to layer requests or nginx for static services like images.

Have you used OpenShift for node apps? If yes, what's your impression?

Azat: No. It's the new hot thing. Eventually everyone will be on some platform as a service, even if it is a private cloud.

Excellent, I've been busy working with PHP

Azat: It's popular because of the number of apps and it's easy to get started. But it has no convention to follow. Creators of Node.js picked JavaScript because of its specs and standards (ECMAScript).

Any Client Side MVC that you would recommend?

Azat: Definitely Backbone and Angular. The answer depends on project: For a highly customized framework with great flexibility in your own framework, use Backbone. Angular is more opinionated, so you must follow their path. If you go to todomvc.com they have 37+ client frameworks and apps for them.

What solution do you use to access documentation when offline?

Azat: Usually I don't. If I have to, I read source codes. There is an app for Mac OS, but I don't use it. I just read source codes. Go to the source file, if node.js, then I would go to node_modules, then press Command F to find the function and see what it is doing. If there is an error, I would clarify by looking at source code.

I want to work more with Node.js.

Azat: Start with Backbone. Just master it. When you become fluent, start by build small apps. Mongo is beautiful, especially with Node.js. You don't have multiple languages to deal. It has one language for all three layers, on the client side and on the database. Meteor is good framework for hackathons, but it does not seem to be popular. It was originally highly opinionated. The wanted to lock you into their own proprietary packaging system. If you learn Meteor, that's all you learn. It's counterintuitive for normal developers, so when you use it, you lose sense with reality.

I like Derby over Meteor.

Azat: I like Derby better than Meteor. It's less opinionated, it has nice templates, and access to database right away. I stopped doing tutorials for Derby because they change it so frequently. MongoUI (<http://github.com/azat-co/mongoui>) written in Node.js and Derby. Go to github.com/azat-co/mongoui or npmjs.org/mongoui. Follow instructions with Read Me.

What are your thoughts on Sencha?

Azat: Sencha is awesome, especially for mobile. Combine with Cordoba to build mobile apps. It works with three applications: iOS, Android, with Blackberry with one source code.

MongoHub?

Azat: MongoHub has UX/UI with tables, not JSON. I tried to do something more intuitive with MongoUI (Not a reliable product yet).

Would reading the ES6 documentation help with understanding JS better?

Azat: Of course, reading any documentation is good, but at some point, you just need to stop and take action. Read a bit, learn a bit. Immediately execute on the learned level. Less effective to read a 200 page doc and then try to effectively implement.

HackHall in chapter 22 in your ProExpress.js book occupied more than 30 pages. Can you talk a little bit more even you had a hold chapter on it?

Azat: It could pretty much a mini-book by itself. It covers a lot of things. Addresses how to build bigger applications than one or two pages.

Do you use Sublime Text? Can you recommend and plug-ins or workflows for it.

Azat: Yes I use Sublime text 2. I used to use TextMe until they dropped the support and made v2 open source. Coolest plug-in = GitHubinator. Open source code in your browser. Workflow = similar to Git workflow, but come up with naming by myself.

What strategy do you use for debugging?

Azat: I'm using console log. 90% of the time the format is not consistent. I can also use node inspector or core debugger. Node inspector is more useful when you don't know what you are looking for.

What's your preferred testing tool/stack/library?

Azat: I use Mocha or Expect. If you need UI testing you could use Casper.js or Selenium. Three basic tests – unit, function and user interface. Unit testing is easy. You don't need an internet connection. Functional tests are more comprehensive. You can use something like Travis CI or CircleCI for continuous integration. For UI, you can use SauceLabs. It will test a bunch of browsers and can record gifs or videos so you can see what is happening.

Is there a good mechanism to estimate what kind of resources will be needed by node app before putting it into production?

Azat: You should do some load testing before releasing. If it's a brand new app, you could write your own small node application or you can search for "node load testing." Apache JMeter is Java-based. It can be launched from server. Usually Node.js is efficient at handling connection and production. Remember to increase you mic sockets value. I have a post about it: Seven Things You Should Stop Doing with JS. Increase the mic sockets from 5 to infinite. Otherwise, it will create a bottle neck for incoming requests.

Do you use Gulp or Grunt?

Azat: For projects like HackHall, I don't use Grunt. Gulp is a little bit faster. I don't use it in books because it will just complicate examples.

Any thoughts on Ember and Ember command line interface?

Azat: It's more mature framework, so go ahead and use it. It is a third interface: Backbone, Angular and Ember. StrongLoop is doing a good job. Its command line interface is getting interesting, approaching Ruby on Rails.

What are the best practices/conventions for Node/Express apps?

Azat: The best practice is to separate routes and keep main file lean and very small. Don't write the whole logic. Copy and paste into a new file. Create a folder (e.g., route/users.js or route/posts.js). If you have a lot of business logic, posts, comments, users that are interrelated, I recommend Mongoose or Waterline. They are object document mappers.

Any thoughts on IO.js?

Azat: I that's awesome. I encourage everyone to use it. It's all open source. It's destined to be great.

Do you use function composition in any of your projects? If so do you find it useful?

Azat: Not sure what function composition is. If you record functions that return functions, that is sometimes called monads.

Do you have thoughts on famo.us?

Azat: Famo.us had lot of hype in the beginning. I have students that used famo.us to build projects at Hack Reactor. They seem to be a bit late coming to the game. Take it with a grain of salt when you hear about something cool that is supposed to revolutionize everything.

Do you have separate repos for front and back end for larger projects or lump them all together?

Azat: It depends. HackHall is all together. Backbone is all together. For a small project you can lump them all together.

Speaking of Hack Reactor (and other boot camps) how useful are they for Intermediate developers?

Azat: I've seen intermediate developers become even better, so you could be advanced, especially if you use PHP and HTML+CSS. Hack Reactor gives you six weeks to build your project. It's pretty useful.

What Node.js books would you recommend?

Azat: You have to get Practical Node.js. This is the one book you should have.