Nate Eagleson
Elizabethtown, PA 17022
717-903-3948
nate@nateeag.com

Full-stack web developer (JS, HTML/(S)CSS, Python, PHP, SQL, bash, Ansible) with sysadmin and devops experience seeking tech lead or senior individual contributor position. Extensive mentoring, code review, and developer tooling experience rooted in a desire to foster effective teams by improving communications and relationships.

Employment

Senior Full-Stack Developer: Nxtbook Media, 2016 - current

I wear many hats beyond full-stack web developer, including dev tooling maintainer, API designer, version control guru (Git, SVN, git-svn, GitHub), code reviewer (GitLab), hybrid mobile app dev, software architect, QA automator, devops lead, and sysadmin. I also occasionally serve as top-tier support tech, business analyst, and technical writer.

I built and maintain an iOS/Android app generation framework that manages ~70 client apps, using Cordova, AngularJS, Angular Material, SCSS, bash, and Node.js. It uses Cordova plugins for in-app purchasing and other features, and I update the Java and Objective-C plugin components as needed.

I wrote an HTTP JSON API to support our mobile app fleet and other systems using PHP, Lumen, Composer, RAML, JSON Schema, MySQL, and nginx. Development Vagrant VMs and production servers are provisioned with the same Ansible playbooks, and we use the API's functional tests for black-box production monitoring.

As sysadmin and devops lead, I wrangle www.nxtbook.com's autoscaling multi-AZ EC2 herd using bash, Ansible, PHP, Node.js, the AWS CLI, and the AWS Console. I added functional tests using bash/bats/curl and use them to monitor production, and added automatic blocking of malicious HTTP bots with a custom fail2ban configuration for our nginx load balancers. I understand what's happening across our EC2 fleet by aggregating logs with syslog-ng and analyzing the logs with standard Unix tools.

Lead JS Developer: United Concordia, 2015 - 2016

I led initial front-end development of BPIR, a system for defining insurance policy benefits for other tools to consume.

I trained and mentored developers ranging from student interns to experienced back-end Java devs in building SPAs with AngularJS, SCSS, Bootstrap, and Lodash, as well as version control and code review with Git. I played a pivotal role in developing front-end team cohesion and camaraderie between team members, turning our motley collection of devs into a working front-end team that got the project done on time and had fun doing it.

I worked with the business and software architecture teams on project planning, requirements gathering, data modeling, API design, specification reviews, and task breakdown and estimation. I coordinated API integration with the back-end team and helped optimize APIs to keep the front-end responsive.

I built and maintained our build/static analysis/test pipeline with Gulp, SCSS, ESLint, Karma, and Jasmine, writing custom ESLint rules and Gulp modules. I implemented the app's core layout, built our screen navigation system with ui-router, and wrote UI components, including a data-driven recursive navigation tree, a JS module to manage the navigation tree's state, a robust auto-saving drag-and-drop editor component, and an abstract tree structure editor.

Front-end JS Developer: MapQuest, 2014 - 2015

As a member of MapQuest's external API team, I worked on tools and APIs available through developer.mapquest.com.

As the JavaScript Maps API's primary maintainer, I added features, wrote interactive code samples with functional Selenium tests (and fought through AOL's infrastructure jungle to get the tests running on all supported browsers), triaged

and fixed bugs, wrote BDD tests with Jasmine, and maintained the API's reference site with YUIDoc. I deployed new versions of the JS API to production and tweaked the Java app for serving the JS module.

I worked with the rest of the team on the Data Manager GUI, a project to build a SPA front-end for our geospatial data management API. We used git for version control, Stash for code review, npm and bower for dependency management, and Grunt.js for running build tasks. We used Mocha and Chai for unit tests, and CasperJS for functional tests. The app itself was written using RequireJS, Backbone, Thorax, Handlebars, Underscore, jQuery, and SCSS.

I also helped teammates with whiteboarding, system design, and brainstorming for various projects, both front-end and back-end.

Full-Stack Web Developer: Clark Associates, 2012 - 2014

I worked on several projects at Clark but mainly on therestaurantstore.com. I wrote new features, triaged bug reports, fixed bugs, wrote docs, reviewed code, designed databases, mentored team members, did code archaeology and resuscitation, handled security scan reports, designed and built APIs, optimized systems, fought wheel reinvention, fixed deployment, and improved developer experience across our team's projects. Our back-end stack was PHP, SQLServer, and IIS, with an HTML / CSS / jQuery front-end supporting Chrome, Firefox, and IE >= 8.

Lead Full-Stack Web Developer: Exponential Gain, 2011 - 2012

I was lead developer, primary code reviewer, sysadmin, and all-around problem solver for initial development and launch of 2x30.com, a website to help people make good habits and break bad ones.

I chose our tech stack (PHP 5.3, Kohana, Twig, Apache 2, MySQL 5, Apache Solr, Ubuntu, git, Gerrit, Pivotal Tracker, jQuery, XHTML, CSS), wrote features, fixed bugs, reviewed code, recruited developers, ran a daily scrum via Skype, helped unstick stuck devs, did technical interviews, and listened carefully to my team.

Freelance Development Services: self-employed, 2009 - 2011

I offered freelance software development and support services, including building custom-branded web kiosks with Webconverger and writing a cross-platform desktop app with wxPython.

Co-founder / Full-Stack Developer: codeninjaz.net, inc, 2007 - 2009

I co-founded codeninjaz.net, inc to build a better way to make website widgets. As one of two founders, I acted as secretary, treasurer, front-end developer, back-end developer, DB admin, and sysadmin. We supported IE 6-7, Safari 3-4, Firefox 2-3, and Opera 9, so I remember the pain of browser compatibility well.

Systems Engineer: Rite Aid, 2005 - 2007

I started at Rite Aid as an Entry Level Systems Engineer working on NexGen, their in-house prescription management system, and earned promotion to Systems Engineer in 2007 by fixing a nasty production bug rapidly. I triaged and fixed bugs, reviewed code, added features, wrote tests, did manual testing, and answered high-level tech support calls.

Education

B.S. Computer Science, Penn State, 2005

Other Interests

audio synthesis – open-source software dev – reading – sustainability – musical instrument design