

Adam Moon

Full Stack Developer

Phone: (214) 600-3604
Email: adammoon.career@gmail.com
Twitter: @jstestabul (https://twitter.com/jstestabul)
Github: adamjmoon (https://github.com/adamjmoon)

About

I am a full stack developer who enjoys working in an agile environment with a collaborative team. I am passionate about creating modular, scalable, smart looking web apps. I enjoy balancing the need for test driven and benchmark driven development, modern creative design, and task completion efficiency and quality. I specialize in .NET and node.js full stack development.

Work Experience

defi SOLUTIONS Inc

Senior Software Engineer
2015 - Present

defi SOLUTIONS provides flexible loan origination system for auto lenders.

- developed several pages using angular, scss, babel & es2015
- implemented a proper js/css build system with gulp
- introduced the use of scss, flexbox, material design, mocha and karma for unit testing, babel for es2015, eslint
- currently working and rearchitecting core app to allow independent js/css/img deployment, and new implementation with react with redux and redux-sagas, react material-ui, and webpack

One Technologies, L.P.

Senior Software Engineer II, Lead Web Developer
2009 - Present

One Technologies is a credit monitoring service company offering such online products as NATIONALCREDITREPORT.com, SCORESENSE.com, and CREDITCHECKPLUS.CO.UK

- Designed and developed Single Page Enrollment Platform to enhance product enrollment experience while allowing full A to B testing. The design focus was small footprint, force js component modular testability, complete design flexibility, and easy path extensibility. Avoided large SPA framewoks to keep size down. Utilized knockout.js, async.js, when.js (promises), and require.js. For path optimization, Only one .css and one .js represented a path utilizing r.js (require.js optimizer). Each path optimization allowed for small footprint of less than 150k gzipped. Created serveral knockout custom bindings and compenents to keep ui component modularity and DOM manipulation separate from business logic. Platform replaced a VB.NET ASP.NET Forms App. I introduced direct automated path deployment to Akamai cdn to allow easy deployment of single path or project including multiple paths.
- Implemented multiple ASP.NET Web APIs in effort to move to Microservice Architecture. Used SimpleInjector for Dependency Injection and Dapper for ORM / Stored Procedures
- Helped develop Single Page Product Portal converting from .NET MVC Web App. Designed to support easy brand extension and flexibility.
- Full stack development on team that created a redesigned Angular 1.4 SPA Customer Care Portal replacing old .NET MVC app. Utilized Google Material Design with the Angular Material directives for UI. Also implemented many fullstack features including Authentication, Role Management, Call Management and Customer Offers and Cancellation.
- Created Productivity Web App for saving, sharing, executing useful SQL Queries for any DB against any environment. Created c# JSONTable Library for automatic dynamic mapping of muliple SQL Query results to a JSON format consumable by frontend for automatic HTML Table generation for Query results. Outstanding performance considering no Type mapping to JSON. Allowed for better experience than SQL Server Management Studio.
- Created Productivity enhancer tool to allow a one command git repos synchronization and bootstrapping of all web app, business apis, and DBs. With a microservice architechture, you have several git repos for individual REST Business APIs which can be a pain to keep in sync with latest code. Utilized node async.js npm module to allow easy parallel processing of each project repos .NET Compilation, DB Creation, Web and Service startup. The dev environment bootstrapping only took as long as the longest individual task. Very freeing result when developers and non developers could keep a clean local env with one command.

Publishing Concepts, L.P.

Software Developer
2007 - 2009

Publishing Concepts provides college alumni print, website, and data services

- Supported and developed on a College Alumni Website product that supported Profile Updating, Alumni Search, Alumni Giving Processing, and Alumni Events Listing and Processing. Single-page site design with multiple .NET user controls. Ex: perunanet.net (SMU alumni site)
- Helped upgrade and convert the Web Project from the .NET 1.1 Framework to the .NET 2.0/3.5 Framework.
- Introduced and Implemented CruiseControl.NET to provide a Continuous Integration Solution for code builds to the Staging Environment. Utilized a combination of CruiseControl.NET built in tasks and NANT scripts to complete the build process.
- Implemented multiple Single Sign-On solutions. Integrated .NET Forms Authentication with multiple SSO authentication protocols. Including LDAP, CAS, Shibboleth, Facebook Auth

Crozier & Henderson Productions Inc.

Software Developer
2005 - 2007

- ASP.NET Template-based website product marketed to homebuilders designed to support multiple builder websites using a shared source code and database. Provided Administration Website to each builder for Content Management and Website Traffic Reporting.
- Developed admin content management pages allowing for CRUD functionalities for builder data and media content
- Prepared system for migration to new data center where apps were to be load balanced across three web servers

Education

Southern Methodist University

Computer Engineering, Bachelor of Science

2001 - 2004

GPA - 3.7

Skills

• Languages

- C#
- Ruby
- C++
- go
- Javascript

• Javascript Precompilers

- Coffeescript
- TypeScript
- ES6 with Babel

• Javascript App Development

- angular
- durandal
- meteor

• Javascript Module Systems

- AMD - require.js
- CommonJS - Node.js, Browserify
- ES6 or ECMAScript Module import
- System.js - supports AMD, UMD, ES6

• Javascript Frameworks

- knockout.js
- backbone.js
- hammer.js
- async.js
- when.js
- lodash.js
- react.js
- react-redux
- redux-sagas
- Immutable.js
- RxJS
- d3

• Javascript Unit Testing / Benchmarking

- frameworks: mocha, jest, jasmine
- mock: sinon
- bdd: chai
- coverage: istanbul
- CI runner: karma
- benchmarkjs

• CSS & HTML

- css3
- less
- scss
- Template Engines: jade, mustache

• UI Automation Frameworks

- selenium
- capybara
- phantomjs
- casperjs

• Build Tools

- nant
- rake
- grunt
- gulp
- jake
- make

• Revision Control Systems

- CVS
- subversion
- git

• Continuous Integration Tools

- CruiseControl
- TeamCity
- Travis
- Bamboo
- Jenkins

• REST APIs

- ASP.NET Web API
- NodeJS: Express, Sails.js

• C# Unit Test

- nunit
- nspec
- Machine.Specifications
- Moq
- RhinoMocks

• IOC and Dependency Injection

- Ninject
- SimpleInjector
- StructureMap
- Autofac

• Database & ORMs

- TSQL
- Stored Procs
- SQL Server 2005,2008,2012
- MongoDB
- NHibernate
- Entity Framework
- Dapper

• Package Managers

- nuget
- webpack
- npm
- bower
- RubyGems

Code Interests

• Languages

- elm
- JS ES201*
- go
- c# 6

• Web Component Library

- angular-material
- material-ui
- material-design-light
- materialize

• App Development

- iOS
- react-native

• Web Component Frameworks

- React
- Angular 2

• Javascript App Architecture

- redux
- redux-sagas
- ImmutableJS

• Javascript Unit Test Frameworks

- mocha, sinon & chai
- ava
- jest
- jasmine

• Javascript Benchmarking

- benchmarkjs
- itchcork

• Focus

- .NET Core 1.0
- Node.js V6
- Isomorphic Apps
- redux
- redux-sagas