NATE GRIGG

nate@nategrigg.com
+1 801 599 NATE

EXPERT SKILLS

Professional Developer since 2002

Nate has expert skills in JAVA 8, WEB SERVICES, C#, and .NET.
But his real passions are in the areas of:
MOBILE STRATEGY, and
SOFTWARE DESIGN and ARCHITECTURE

PROFICIENCIES

While he wouldn't claim to be an expert, Nate is proficient in REST/HATEOS, JAX-RS, .NET CORE, ASP.NET MVC, MAVEN, ECLIPSE, INTELLIJ IDEA, VISUAL STUDIO, NUGET, MICROSERVICES, SCRUM, and LEAN DEVELOPMENT

EMPLOYMENT HISTORY

MAGiE - Personal Project - Mobile Puzzle Game

Nov 2018–Present

magiegame.com

C# 7, .NET CORE on Mac, NUGET, COMPONENT-ORIENTED DESIGN, APPLE APP STORE, GOOGLE PLAY STORE

Senior Software Engineer, Ancestry Inc.

Apr 2015–Dec 2018

JAVA 8, MAVEN, JUNIT, JAX-RS, AWS, REST,
MICROSERVICES, (Including breaking up a monolith),

REFACTORING, AUTOMATED TESTING, JENKINS, SCRUM

Senior Software Engineer, Intermountain Healthcare

Mar 2014–Apr 2015

C#, ASP.NET MVC, SITECORE, REST, SCRUM

Senior Software Engineer, RedBell Real Estate

Sep 2013–Feb 2014

C#, ASP.NET WEB FORMS, MS SQL, STORED PROCEDURES

Consultant → Senior Consultant, Application Development, Avanade Inc.

May 2006–Sep 2013

C# 2.0–5.0, ASP.NET MVC, WEB SERVICES, WCF, MS SQL SERVER

Software Engineer in Test, Control4

2004–2006

LINUX, EMBEDED, C#, C++, SQL SERVER, XML, WIFI, ZIGBEE, TEST PLANNING, AUTOMATED TESTING, MANUAL TESTING

Software Developer, ThoughtLab

2002–2004

ASP.NET WEB FORMS, MS SQL SERVER

EDUCATION

Bachelor of Science, Computer Science
University of Utah

2002–2006

Also Attended

Utah State University, 1997

Salt Lake Community College, 1996–1998

PROJECT HIGHLIGHTS

Monolith Break-Up and conversion to Micro Services

Ancestry 2016–2018

The Commerce department at Ancestry was already working on replacing a few large MONOLITHS with smaller services. They were using that project as a springboard to be one of the first departments at Ancestry to switch from .NET to JAVA. Nate and some other colleagues joined Commerce all at once. This overloaded the CODE REVIEW WORKFLOW. Specifically, it introduced a lot of merge conflicts due to simultaneous work being done by many developers.

One the solution was to break up one of the new Java services into three *even smaller* micro services. This reduced merge conflicts from unrelated code changes by moving unrelated code into different code-bases.

At the same time, the team worked to "lift and shift" all of these services: the MONOLITHIC and the MICRO services, as well as the in-between "MINI" services.

All this while implementing new features, raising standards for software excellence, and improving delivery times for new features.

Java 8, JAX-RS, REST, Micro-Services, Spring Dependency Injection, Maven

Mobile Web Application for booking flights

2010–2012

Nate built a prototype MOBILE WEB APPLICATION for making airline reservations using *Navitaire's* NEW SKIES reservation system. One of Navitaire's major European customers needed a mobile booking flow *yesterday*. Nate proposed polishing the prototype web app and the customer agreed.

During the polishing project, the CEO of the Airline insisted that the mobile booking flow be accessible from ANDROID and IOS *apps* in the app stores. While development continued on the mobile web app, developers working for the airline built native apps with web controls to wrap the mobile web app.

Patent for "One-Click Booking" for A Bus Ride

2010–2011

US Patent 8942991B2: Agent-side traveler application for mobile computing devices
While making a proof-of-concept for a customer pitch, Nate and his colleagues at *Navitaire* made a web app that would allow a bus driver to reserve and sell a seat on the bus with a single swipe of a credit card. Accenture decided to patent it.

Long story short: the app already knows the route based on driver schedule, GPS, or some other input. So the app already knows the destination. The bus driver tells the app which stop they are at, (or it figures it out from GPS readings) so the app knows the origin of the reservation. Buses don't have stringent identification requirements, so passenger information does not need to be collected at the time of sale. So all the driver has to do is take the money and the reservation is made.