

NATE GRIGG

nate@nategrigg.com

+1 801 599 NATE

I have 20 years of development experience, and an intuitive creativity that is hard to come by in engineers. Combining my creativity with an uncanny knack for abstraction leads me to ingenious technical solutions. Mixing in my natural aversion to inefficiency brings us to truly elegant designs.

SKILLS/TECHNOLOGIES

- **DIGITAL/MOBILE** Strategy (See [Mobile Projects](#) below)
- **DESIGN** and **IMPLEMENTATION** of **APPLICATIONS** and **WEB SERVICES**
- **MICROSERVICES**
- **REST WEB SERVICES**
- **JAVA**
- **JAX-RS**
- **SPRING**
- **PYTHON**
- **DJANGO**
- **FLASK**
- **.NET**
- **ASP.NET MVC**
- **.NET CORE**
- **LEAN DEVELOPMENT**

WORK HISTORY

Senior Software Engineer, Merit International Inc.

May 2022–August 2022

GO, KAFKA, POSTGRESQL, GOOGLE CLOUD PLATFORM

Implementing features in back-end microservices

Senior Software Engineer, ReliaQuest

August 2021 – March 2022

JAVA microservices built with **SPRING BOOT**, **SPRING MVC**, and **POSTGRESQL** (deployed to **AWS**).

I was the “Squad Lead” for a full-stack team: me and 2 back-end developers, 2 front-end developers, a UX designer, and a product owner.

The project was to port an internal, employee-facing application to be part of ReliaQuest’s main customer application, “GreyMatter”.

We coordinated with ReliaQuest’s Data Engineering team to set up jobs to transfer the data from the active internal app into a **POSTGRESQL** database. We also worked with the **DEVOPS** team to spin up a brand-new **MICROSERVICE** (developed by the other back-end developers and me) to serve that data to the front end.

The front-end developers on the team built reusable components to display the information to the users of the GreyMatter web app. (Building one of these components for temporary use is my only **REACT** experience.)

We were coordinating beta testing when I left the company.

Principal Software Engineer, Newfold Digital

A.K.A. Bluehost

Dec 2019 – Aug 2021

PYTHON, **DJANGO**, **FLASK**, **MICROSERVICES**, **JSON WEB TOKENS (JWT)** I was the Tech Lead on a team with four fellow developers, an engineering manager who coded, and a test engineer.

We built a **DJANGO** microservice from scratch and inherited two **FLASK** microservices.

Content Recommendation System

From scratch, we built a **REST MICROSERVICE** in **DJANGO** to recommend helpful links (such as knowledge-base articles) to customers at various page locations in Bluehost’s web apps.

The service was so helpful, easy to use, and robust, that the marketing team decided to use it to show **BLACK-FRIDAY OFFERS** to new and existing customers (in both the U.S. and an international market that represented a lot of growth potential for the company).

Website Builder Authorization Service

One of the **FLASK** microservice that we inherited signed **JSON WEB TOKENS (JWT)** to authorize users of Bluehost’s new Website Builder for WordPress sites. The service was already running in beta and production; we continued implementing and enhancing it.

Website Builder API Gateway

We also inherited another **FLASK** microservice that routed calls from the new builder back to [Bluehost's](#) monolithic Hosting Platform.

Developer, Personal Project

Nov 2018–Oct 2019, Ongoing

- Built a puzzle app built with the **UNITY** game engine
- Released it to Apple's iOS App Store
- Released it to Google's Play Store

See [MAGiE](#) below

Senior Software Engineer, Ancestry, Inc.

2015–2018

JAVA 8, JAX-RS, REST, MICROSERVICES, SPRING DEPENDENCY INJECTION, MAVEN

2016–2018 Monolith BreakUp and conversion to Microservices

Commerce department at Ancestry

The whole department was working on replacing a suite of monolithic **.NET** services with Java “**MINI**services”. Aiming to build **MICRO**services, the team ended up building services that were a little larger than intended.

So, while the larger team worked on implementing new features and breaking up the **MONOLITH**, a smaller team (including me) also worked on breaking the **MINI**services into **MICRO**services. The department also spent the last year “lifting and shifting” all of these services (**MICRO**, **MINI**, and **MONOLITHIC**) to the **AWS** cloud.

All this while implementing new features and helping the team raise their standards for software excellence. I have written up some case studies about some of this work - *ask me for details!*

2015–2016 Cross-cut Solutions Team

Audited Ancestry's services catalog and assisted teams to integrate their services with the proprietary **APPLICATION PERFORMANCE MONITORING (APM)** solution.

Converted a **.NET MICROSERVICE** to **JAVA 8**

Senior Software Engineer, Intermountain Healthcare

2014–2015

Non-Profit Healthcare Provider, regional hospital network.

One of two back-end developers on the web team implementing the company's main user-facing web site.

See [REST endpoint...](#) in [Mobile Projects](#) below.

SITECORE CONTENT MANAGEMENT SYSTEM (CMS)

Senior Software Engineer, RedBell Real Estate

2013–2014

Added new features to an existing web application for a real-estate services company

ASP.NET WEB FORMS, MICROSOFT SQL SERVER, STORED PROCEDURES

Senior Consultant, Application Development, Avanade Inc.

2006–2013

IT Consulting joint venture from Accenture and Microsoft

2013 National Retail Pharmacy

I worked with a national retail pharmacy to integrate what they thought was an abandoned scheduling tool into their new consumer web application. The developers who had built the appointment-scheduling application had all left the company, and the appointment-scheduling in the new app wasn't ready.

I was able to assess the old appointment tool and modify it so it could be shown as an *iframe* in the new web app.

2006–2013 Navitaire Professional Services

I spent many years working with many airline clients with Navitaire's Professional Services group.

Most of the work was advising web developers at the various airlines how to implement custom features in Navitaire's ASP.NET web application.

See: [Mobile Web Application for booking flights](#) and [US Patent 8942991B2](#) below.

Software Developer in Test, Control4

2004–2006

Home-automation startup, paid internship

I was the first to run Control4's main dashboard application, "Media Controller", on Control4 hardware. We had been running the controller on commodity PC's when the first batch of prototype hardware arrived. I saw that stack of machines and said, "might as well give it a go!"

So I grabbed one of the machines off the pallet and installed the OS and applications.

I also built an application to help a subject-matter-expert test our database of remote-control codes.

Embedded Linux, C, C++, Custom Test Automation Software in C#

Software Engineer, ThoughtLab

2002–2004

Custom web applications using the Lean Development methodology

I immensely enjoyed an on-site visit to one of our customers: a furniture-manufacturing company. Visiting with the people at the company taught us so much about their business and their needs. We really gave them the best features within the project schedule.

ASP.NET 1.0, 1.1

MOBILE PROJECTS

MAGiE: MAGnetic Interactive Explorer

Nov 2018–Present (When free-time allows)

I released a mobile game, built using the UNITY game engine with **C#** scripting. While implementing it, I built a small library for the underlying bit manipulation and used NUGET to manage that dependency in the UNITY project. Working on my Mac, I learned about DOTNET CORE and the **JETBRAINS RIDER** IDE to make life easier.

While managing beta tests and releases, I earned a new respect for continuous delivery and release management!

The game is called MAGiE, short for MAGnetic Interactive Explorer. The premise is that you have this device that reads binary data from a mag-stripe like on a credit card. The game then presents you with puzzles where you have to decode the bits to read a message, or encode bits to answer a puzzle.

It's lots of fun, you should try it!

<https://magiegame.com/magie/>

REST endpoint to provide mobile app with content from Sitecore Content Management System

2014

While staffing a project to convert the main website from **SHAREPOINT 2010** to the **SITECORE CONTENT MANAGEMENT SYSTEM (CMS)**, a new feature was being developed in the company's native mobile apps. I quickly exposed content from the **CMS** as **JSON** output so the mobile developers could deliver their feature on time.

Mobile Web Application for booking flights

2010–2012

I had 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*. I proposed polishing the prototype web app and the customer agreed.

I led a revolving team of developers to polish the web app for production use at a major airline. As the project was ramping up, the CEO of the airline insisted that the mobile booking flow be accessible from **ANDROID** and **IOS apps** in the app stores.

As we continued developing the mobile web app, I worked with developers at the airline who built native apps with web controls to wrap the mobile web app.

US Patent 8942991B2: Agent-side traveler application for mobile computing devices

Application Filed May 2011

This patent illustrates my ability to reduce complexity. Here, the steps of finding a bus route, selecting a destination, booking a ticket, and collecting payment are reduced to a single swipe of a credit card.

Mobile Game in the Windows Phone marketplace

2011

My buddy filled the game-designer role, and I developed and released a puzzle game to the **WINDOWS PHONE** marketplace. I found many exciting technical challenges including programming animations and subtle interactions in the **WINDOWS PHONE** flavor of **SILVERLIGHT**.

Releasing the **MINIMUM VIABLE PRODUCT** taught us a lot about **PROJECT MANAGEMENT** and **PRIORITIZATION**. While we had hoped for greater success for the game, the lessons learned were well worth the effort.

EDUCATION

Bachelor of Science, Computer Science University of Utah

2002–2006