

# BARRY HAYLES

---

Full-Stack Software Engineer

---

CONTACT 817-819-5588 | [devdoctor@live.com](mailto:devdoctor@live.com) | Pembroke Pines, FL

---

SKILLS

- C# .NET
- NodeJS
- Python
- C++
- SQL Server
- MongoDB
- Snowflake
- Test Driven Development
- Angular
- ReactJS
- VueJS
- Data Visualization
- Machine Learning
- Problem Solving
- CI/CD
- Cloud Development

---

EDUCATION New York University | BS Computer Science | GPA 4.0

---

## EXPERIENCE

Franklin Templeton Investments, Fort Lauderdale

2015-2025

Manager / Senior Engineer

I started as a senior developer and worked my way up to lead, and eventually manager of a team of 6 developers, to support our Fixed Income Investment Management teams, in areas of both Risk and Performance analysis. As a hands-on manager, I successfully lead a team to create and maintain several applications to automate and manage compliance; reporting and visualizations... I and my team, designed and developed solutions that saved the department hundreds of man hours per month.

### C# .NET:

- I created .NET Core middleware applications to automate scheduled task orchestration, and provide access to catered data, sourced from various databases; rest APIs; CSV and Excel files; FTP Servers; ...
- I'm proficient with both MVC and Minimal API
- I utilized IOC dependency injection, with proper lifetime management (transient; scoped; singleton; factories)

- I utilized Test driven development using XUnit to test middleware components (using Fixtures)
- I'm proficient using TPL asynchronous programming (tasks, dataflow, parallelism, async / await ...)
- I used Reactive extensions for data aggregation logic
- I used MSAL / OAuth2 to secure endpoint access and authorization
- I used Docker containers, deployed to both MS and Linux, Azure App Services from Azure Container Repository
- I'm proficient with Azure DevOps: Work Items; GIT Repositories; CI/CD Pipelines; and Artifacts (for an internal Nuget server.
- I designed complex build and release pipelines
- I modernized many .NET Framework applications to Core
- I utilized Entity Framework often to build data access layers for REST APIs
- I utilized SignalR for server to browser event messaging; streaming and notifications

#### NodeJS:

- I created middleware applications in Typescript using ExpressJS to serve augmented data for downstream reporting
- I implemented RESTful API, including implementing OData 4.0 endpoints for Excel and PowerBI access to data
- I designed and developed a custom IOC dependency injection framework to use with Express route handlers
- I developed an OAuth2 middleware to secure endpoints
- I automated endpoint routing registration, that utilized Typescript decorators.
- I utilized RxJS to optimize data aggregation logic.
- I heavily utilized async / await, Promise based methods in route handlers.
- I utilized Node clustering for endpoint handling, scaling and asynchronous task processing
- I created Docker and Kubernetes deployment in Azure App Services using Azure DevOps

#### CI/CD:

- I used modules (like GULP) to create build and deployment scripts
- I used Mocha and Chai to build unit tests, and integrated the test into the build process

#### Python:

- I created middleware using Flask to host SCC compliance reporting applications
- I utilized Python libraries to access numerous types of data stores (including SQL; Oracle; Snowflake; MongoDB)
- I used Anaconda and various ML packages (like Numpy and SciPy; ...) to train and augment Fixed Income RISK statistical data, stored in a Snowflake data warehouse

- I utilized Python scripts to improve build and deployment processes
- I used Anaconda and Jupyter Notebooks to prototype and visualize statistics and RISK analysis data

### SQL Server:

- I lead design and development of database schemas
- I headed database normalization and performance improvements
- I utilized and dynamically managed SQL Jobs and Schedules for automating tasks
- I used Service Broker as a middleware queuing system to provide transactional (ATOMIC) operations and automate orchestrated tasks.
- I implemented concurrent processing by utilizing queues and stored procedure activations
- I created scripts for highly performant reporting, using common table expressions (CTE); extensive aggregations; and window functions
- I created ETL processes to extract; clean; normalize and store data ensuring data integrity (we want clean data in – clean data out)
- I was heavily involved in performance tuning and monitoring
- I designed and implemented temporal tables; graph tables (with queries); and partitioning

### MongoDB:

- I designed databases schemas; views with aggregate pipelines; reports and visualizations for financial securities and holdings information
- I created and fine-tuned aggregation pipelines to provide reporting data (utilizing lookups and graph lookups... among other aggregation stages)
- I created MongoDB Charts, to visualize and analyze financial securities data stored in MongoDB Atlas
- I implemented MongoDB Shell scripts, used to automate analysis; reporting and deployment processes
- I maintained database optimization, including suggesting and or applying index recommendation
- I designed graph databases

### Angular:

- I implemented most web applications using Angular and Bootstrap CSS library
- I modernized legacy Angular application to use latest version of Angular (Ivy)
- I lead the design and implementation of Angular applications and libraries in multi-project solutions
- I utilized Angular Environments for SDLC deployments
- I reengineered component state management to utilize service-based state management

- I reduced “prop-drilling” using service-based state management, to improve implementation; maintenance; and performance
- I utilized Angular Signal Store (in replace of Common Module constructs and state management) to improve performance
- I implemented custom HTTP interceptors to handle authentication and authorization from REST API responses
- I designed and implemented custom Pipes used by components that render dynamic components (for dashboards; tabular navigation; ...)
- I capitalized on code reuse by implementing abstract components and services...
- I used Routing to build SPA applications
- I utilized NgRx to manage global state management
- I implemented and utilized unit testing using Karma, for Pipes; Services; Components; ...

### ReactJS:

- I created a few ReactJS web applications for POC
- I developed customer facing application used for marketing; signup; profile management; ...
- I implemented both class and (mostly) function components
- I utilized several hooks (useMemo(); useState(); useEffect(); useContext(); ...)
- I used useContext(); useMemo(); and ReduxJS to manage global and context state, to improve performance by reducing “prop drilling” and component access to state
- I created “clean” JSX code by breaking up components into multiple sub-components

### Cloud AZ/AWS:

- I’m proficient in managing all features of Azure Entra ID, including App Registrations, secrets, API access...
- I utilized Key Vaults to store sensitive information used for authentication
- I’ve deployed and managed SQL SaaS servers
- I’ve created Functional Apps to that used Cosmos DB to quickly fetch minimal financial data
- I’ve utilized Blob Storage to serve media files (e.g. videos; images; ...) using both Azure Storage and AWS S3
- I utilized Azure Service Bus to send and receive high throughput Equity data, for aggregation processing and storage
- I used both the NodeJS and .NET libraries to administer and monitor Azure resources

### Data Visualization:

- I created visualizations (reports and charts) using PowerBI
- Used Power Query to import and create transformations and views of Fixed Income Risk exposure data models
- I utilized Pivots and Unpivots; Window Functions; Aggregations

- I'm proficient in DAX for creating measures; conditional columns; creating relations between fact and dimension tables; and data filtering
- I utilized most visualizations to create bar; line; pie; and clustered charts

#### Miscellaneous:

- I effectively communicated design specifications
- I lead, design and Agile / SCRUM sessions
- I performed code reviews to ensure proper programming practices; security and bug mitigation; and coding standards
- I built vision and influenced team members; other managers and directors; and associates to develop enterprise-level and innovative business application
- I lead Disaster Recovery (DR) design; implementation; and sessions
- I established proper software development lifecycles (SDLC) environments, and GIT Repository practices
- I provided leadership and mentoring to my team and provided accountability for performance; usability; and quality in our applications
- I consistently directed and shared improvements in processes and best practices
- I'm proficient in developing and verifying software requirements
- I lead the hiring processes (interviewing and candidate selection)
- I mentored all (including many Junior and Mid-level developers), allowing them to become very successful engineers
- I'm fluent in architecture and design patterns
- I have strong problem solving and debugging skills
- I'm a quick learner and excellent teacher
- I have extensive knowledge of development tools; APIs; packages; ... across several technologies and environments
- I created visualizations using MongoDB Charts
- I'm extremely proficient in HTML; CSS; and Java/TypeScript. I used much of these to implement creative; responsive; and innovative user interfaces
- I created a few web applications using VueJS
- I solely designed and developed a Self-Serve automation system that allowed risk teams create, manage and to schedule task, such as downloading files from FTP servers; extracting and creating ZIP files; convert Excel Workbooks into data sets; query data sources; create reports in CSV and Excel; run Excel macros; ... Many of these tasks are data and CPU intensive, literally crippling their workstations. This saved 10's of thousands of man hours annually, enabling efficiency and relieving risk teams from performing these tasks manually, by offloading these tasks to powerful servers

## Infusion Development

2011-2015

### Senior Engineer Consultant

Led the development of a high performance, high throughput, and low latency FOREX trading system for THE CUREX GROUP, NYC. I was primarily tasked in implementing the liquidity and execution gateways, using FIX protocol libraries in C++. I was also involved in implementing the logic in the matching engines. I lead the development of Win32, low level asynchronous processes that processed orders; credit and liquidity using OS level thread pools; no-lock constructs; parallelism; ...

#### C++ (v20):

- I created a FIX high speed (read only) parsing library using Boost; PPL; STL string and unordered\_map, that pulled FIX data from Berkely Sockets to build protocol buffer classes.
- I serialized and deserialized queue messages from a high-performance Solace queuing middleware (using Protocol Buffers)
- I utilized lambdas and STL library components...
- This includes shared\_ptr, unique\_ptr; atomic; ...
- Heavy use of OS level interlock functions for lockless asynchronous programming
- Heavy use of header only templates and classes
- I implemented high performance Berkely Socket (TCP/UDP) class for FIX protocol input and output
- I utilized Azure DevOps GIT repository for source control
- I participated in SCRUM meetings for project planning and iteration management
- I provided guidance and consultation in the proper way to use GIT

#### SQL Server:

- I successfully redesigned and implemented a more effective database schema (tables; views; stored procedures; functions) to manage orders and archive liquidity data
  - I created multi-dimensional models (cubes) in Analysis Services build a business intelligence reporting system to data mine the liquidity data.
  - I created cubes and SQL MDX queries to discover patterns and trends, present in the data.
  - I created predictive models, many of which were used, to provide insights to the floor traders
-