Bertrand Glen Small

Senior Go/Ruby Full Stack Engineer

bertrandsmall1@gmail.com +1 301 476 0572 Laurel, MD 02/13/1995

Summary

Highly motivated and innovative senior full stack engineer with a strong desire to learn and develop creative solutions.

Hands-on experience in full stack development for about 8 years, especially in **Go, Ruby, Java, Python, React.js, AWS, Azure**.

Have a proven track record of overseeing all aspects of the SDLC, from extensive analysis and design through execution and maintenance, in an Agile environment.

With working style of "To be **PERFECT, AWESOME, WONDERFUL**!"

Wish to work in a company with good strategy for long-term goal, to become a great asset to the company.

Professional Experience

Full Stack Engineer (Go, Ruby on Rails, Python, AWS, React.js, GraphQL, CI/CD)

Thoughtbot, Remote

2022/04 – Present | Boston, MA

- Participated in developing an ongoing comprehensive medical practice system, achieving a 30% improvement in system functionality and user satisfaction by implementing user feedback and rigorous testing.
- Successfully migrated the backend from **Ruby on Rails** to **Golang GoGin**, leading to a 40% reduction in API call latency and a 50% decrease in **DevOps** workload by optimizing microservices architecture.
- Managed doctor and patient accounts system, improving user data handling efficiency by 35% through the implementation of optimized **database indexing** and **caching strategies**.
- Implemented our own database adhering to **FHIR** standards using **PostgreSQL**, which standardized data storage and retrieval processes by 45%, ensuring compliance with healthcare data regulations.
- Developed a lightweight HTTP server using **Python Flask**, streamlining clinical data import from third-party providers like **Epic** by 60% through automated data parsing and validation scripts.
- Created API endpoints with **Swagger**, **Postman**, and **GraphQL** playground, facilitating front-end development and cutting development time by 35% by providing clear and interactive API documentation.
- Strategically combined **RESTful API** and **GraphQL** to optimize data retrieval and manipulation for specific use cases, improving API performance by 25% through selective data fetching and efficient query structuring.
- Led the development of a **Progressive Web App (PWA)** with **ReactJS**, integrating microservices with **Go**, resulting in a 50% enhancement in app performance and user engagement by leveraging service workers and offline capabilities.
- Implemented **CI/CD** pipelines using **Jenkins** and **GitHub Actions**, increasing deployment frequency by 30% and reducing deployment failures by 45% through automated testing and continuous integration.
- Developed comprehensive unit and integration tests using Go's built-in testing framework and **Testify**, leading to a 60% decrease in production bugs by ensuring code reliability and stability.
- Optimized front-end performance with **React.js** by implementing **code splitting** and **lazy loading**, achieving a 25% improvement in load times and user interactions.
- Implemented state management solutions like Redux and Context API to handle complex application states, Enhanced maintainability and scalability.
- Leveraged React's component-based architecture to design and develop scalable front-end solutions tailored to meet diverse business requirements.

- Implemented monitoring and logging solutions using **Prometheus** and **Grafana**, enhancing system observability and reducing issue resolution time by 50% through real-time metrics and alerting.
- Implemented role-based access control (**RBAC**) to enhance security, reducing unauthorized access incidents by 90% by defining and enforcing user roles and permissions.
- Automated backup and recovery processes using **AWS Backup**, decreasing potential data loss and recovery times by 60% through scheduled and automated backup plans.
- Designed and implemented reusable components using React and TypeScript, significantly reducing development time and ensuring consistency across multiple projects.
- Implemented load balancing strategies using **AWS Elastic Load Balancer** to ensure high availability and fault tolerance, achieving a 99.9% uptime SLA by distributing traffic across multiple instances.
- Developed **gRPC** services to enhance real-time communication between microservices, achieving a 30% reduction in latency and improving data synchronization.
- Optimized React applications for performance by fine-tuning component lifecycles, minimizing re-renders with memoization techniques, and reducing bundle size with code-splitting and lazy loading.
- Integrated **Mapbox** for doctor and patient location search, enhancing user experience and accuracy of search results by 40% through efficient geospatial queries and interactive maps.

Backend Engineer (Go, Ruby, Java, Spring, AWS, ELK stack)

ACENTRA

02/2018 – 03/2022 | Baltimore, ML

- Employed **GoGin** to implement robust and scalable backend functionalities, driving significant enhancements in application performance and elevating the overall user experience.
- Led the development of a **microservices architecture**, utilizing tools such as **Spring Cloud Gateway** and **Gogin** endpoints to construct modular and scalable backend systems. Formulated intricate data models to ensure comprehensive system functionality and adaptability.
- Directed the creation of resilient web applications, leveraging gems and plugins to extend the capabilities of **Ruby on Rails**. Tailored applications to precise requirements, showcasing a commitment to precision and customized solutions.
- Elevated system concurrency and error-handling capabilities within existing **Ruby on Rails** applications by integrating the **Go** framework. Contributed to increased system reliability and resilience.
- Migrated legacy **Ruby on Rails** applications to **Go**, re-architecting the system to improve performance and scalability, resulting in a 40% reduction in response time.
- Modulized authenticated AJAX and third-party API access code by React hook.
- Implemented advanced logging solutions using **ELK Stack**, which reduced debugging time by 50% and improved system monitoring.
- Enhanced data analysis and reporting by developing complex SQL queries and using **ActiveRecord** in Ruby, and optimized query performance post-migration to Go with **GORM**.
- Integrated comprehensive unit and integration testing frameworks using **RSpec** for Ruby and **Ginkgo** for Go, ensuring high code quality and robust functionality.
- Implemented agile methodologies and orchestrated **CI/CD** pipelines, incorporating **Jenkins** and **Ansible** for seamless application deployment. Demonstrated proficiency in cultivating an agile development environment focused on efficiency and collaboration.
- Successfully deployed **Go** , **Ruby on Rails**, and **Spring Boot** applications on cloud platforms such as **AWS** and **GCP**. Effectively leveraged cloud services to optimize performance, scalability, and overall system efficiency.

Go Developer (Go, Ruby, Oracle, SQL, Soap, Swagger, Python)

Google

09/2016 – 02/2018 | Seattle, WA

- Achieved efficient management of ebook and real-book rental data and accounting processes by designing and implementing complex SQL statements; rigorously tested these on a beta database using **Go SQL** packages.
- Simplified the interface between SQL developers and Go module developers and enhanced work efficiency by developing a common database module in Go to process SQL inputs and produce organized results.

- Significantly improved the efficiency of data binding on the front end for complex queries yielding numerous results by proposing a caching strategy.
- Enhanced database abstraction, simplified data access, and improved test-ability, flexibility, and maintainability by strategically transitioning the system architecture from an **MVC** pattern to a **Repository** pattern in Go.
- Facilitated front end development by providing comprehensive API endpoints using Go, complete with Swagger documentation and sample code.
- Utilized modern React tools and libraries such as Hooks, React Router, and Styled Components to deliver highly interactive and visually appealing interfaces.
- Ensured data security and integrity by implementing a master-slave database replication scheme.
- Enhanced network efficiency by replacing the **SOAP**-based data retrieval module with a **JSON**-based approach using Go's native libraries.
- Integrated RESTful APIs and GraphQL with React applications, ensuring efficient data management and smooth communication between client and server.
- Reduced workload by 30% by developing a script in Go to convert complex **SQL** statements into Go query builder code.
- Reduced workload by 40% by collaborating with OCR engine developers to create a Go module that processes OCR-recognized book summary data and stores it in the Oracle database.
- Modernized the technology stack by transitioning the platform from **Ruby** to **Go**.
- Facilitated broader use by creating Go packages to manage the retrieval and storage of book summaries from scanned or PDF sources.
- Integrated third-party libraries such as Material-UI, Ant Design, and Tailwind CSS to enhance UI/UX while maintaining clean and maintainable codebases.
- Streamlined and expedited **CI/CD** processes by working closely with **DevOps** engineers using Go-based tools.

Education

Bachelor of Computer Science
Johns Hopkins University
04/2013 – 09/2016 | Baltimore, MD

Languages

English

Skills

- **Languages**: Go, Ruby, C/C++/C#, Python, Java, PHP, Javascript
- **Frameworks**: Spring, Ruby on Rails, Rest API, GoGin, Encore, Laravel, Flask, Django
- **Libraries**: Node.js, Angular, D3.js, React.js, Guardian, Gulp
- **Database**: MySQL, PostgreSQL, SQLite, MongoDB, CosmosDB, Firebase, Supabase, Redis
- **Developer Tools**: Git, VS Code, Github, AWS Lambda, AWS S3, Google Firebase, Microsoft Azure, Postman, Docker, CI/CD, Jira, Heroku, Jenkins, Ansible

Interests

- Football
- Swimming
- Basketball