

# Fred Zhang

Winnipeg, MB | +1 (204) 770-3168 | fredzhang026@gmail.com

Portfolio: fredz6.github.io/portfolio | GitHub: github.com/FredZ6 | LinkedIn: linkedin.com/in/haifeng-zhang26

## SUMMARY

Java Backend Developer focused on Spring Boot and production-style distributed systems. Built event-driven microservices with Outbox, idempotency, retry/DLQ, observability, and automated delivery on AWS. Seeking Backend Software Developer roles.

## EDUCATION

Bachelor of Science in Computer Science (Minor in Mathematics), University of Manitoba Sep 2019 - Apr 2024

## SKILLS & CERTIFICATIONS

Backend: Java 17, Spring Boot 3.x, Spring Security, REST APIs, JWT, RBAC, Maven, OpenAPI/Swagger

Distributed Systems: Event-driven architecture, Outbox pattern, idempotent APIs/consumers, retry & DLQ, eventual consistency (Saga-style compensation)

Data & Messaging: PostgreSQL, Redis, RabbitMQ

Observability: OpenTelemetry, Jaeger, Prometheus, Grafana, Alertmanager

Testing: JUnit 5, Mockito, Testcontainers, OpenAPI contract tests

Cloud/DevOps: Docker, Docker Compose, Git, GitHub Actions, Terraform, AWS (ECS, ECR, CloudWatch), GitHub OIDC

Certifications: AWS Certified Developer - Associate (DVA-C02); AWS Certified Cloud Practitioner (CLF-C02)

## PROJECTS

**Event-Driven Order & Inventory Microservices Platform - GitHub: <https://github.com/FredZ6/cloud-project>**

*Java 17, Spring Boot, RabbitMQ, PostgreSQL, Redis, OpenAPI, OTel/Jaeger, Prometheus/Grafana, Terraform*

- Built an event-driven microservices backend (Auth/Catalog/Order/Inventory/Payment/Notification) orchestrating order-to-payment flows via RabbitMQ topics.
- Implemented Outbox pattern in order-service (outbox\_events + scheduled publisher) to guarantee reliable event publishing after DB commits.
- Added end-to-end idempotency: API idempotency for POST /orders via required Idempotency-Key, plus consumer deduplication (consumed\_messages) across inventory/payment/order services.
- Engineered production-style resilience with TTL retry queues + dead-letter exchanges and per-service \*.dlq routing after maxRetries.
- Delivered full observability: OpenTelemetry tracing (Jaeger), metrics (Prometheus/Grafana), and alerting (Alertmanager), plus health probes and runbooks for diagnosis.

**E-Commerce Platform - GitHub: <https://github.com/FredZ6/e-commerce>**

*Spring Boot 3.x, React 18, PostgreSQL, JWT + RBAC, Docker Compose, GitHub Actions, JUnit/Vitest/Playwright*

- Delivered a full-stack commerce app with customer + admin workflows, packaged as a multi-service Docker Compose stack for one-command demos.
- Built quality automation with 39 tests (29 backend JUnit + 10 frontend/E2E via Vitest/Playwright), GitHub Actions pipelines (CI/E2E/deploy/destroy), enforced coverage gates, and OpenAPI-based contract tests for API compatibility.
- Hardened security with secret validation, login abuse protection (rate-limit + block window), and admin-only protection for privileged/actuator endpoints.
- Published a reproducible performance baseline (200 requests, concurrency=10): GET /api/products P95 21ms @ 88 rps.

## EXPERIENCE

**Software Engineering Intern | TechNovate Solutions | Jiangmen, China Jun 2022 - Aug 2022**

- Built a customer analytics dashboard using React + Flask, reducing manual reporting effort by ~25% (estimated via report cycle time).
- Designed and integrated RESTful APIs across SaaS modules (e.g., payment/CRM), improving p95 latency by ~18% via caching and query optimizations.
- Collaborated in Agile sprints, authored onboarding and technical docs, and improved handoff quality for subsequent contributors.

**Finance Assistant | Southland Restaurant | Winnipeg, MB Nov 2019 - Jun 2025**

- Generated monthly/annual financial reports with SQL, supporting cost tracking and budgeting for 3 departments and 20+ vendors.
- Built Python forecasting/variance tools, improving accuracy by 12% and reducing report preparation time from 6.0h to 3.5h.
- Automated payroll processing workflows with Python, reducing calculation errors by ~30% and improving audit readiness.
- Reconciled supplier transactions using advanced SQL; identified anomalies worth ~CAD 6,000/year.