Helian Peng (David)



Java Backend Engineer
2026 graduate with an Alibaba Cloud private-cloud security backend internship; focused on Java services, reliable tasks, Redis concurrency and production diagnosis.
Chongqing
Recruiters interested in my experience are welcome to contact me by email. ·Email · lysf15520112973@163.com ·GitHub ·DavidHLP ·Knowledge base

Selected Strengths

- **Reliable tasks**: Used Outbox, Redis Streams and SQL CAS for UltiCode delivery gaps, duplicates and rejudge races; added real-MySQL and browser/Judge acceptance to CI.
- **Caching and concurrency**: Built same-key request coalescing, distributed loading coordination, version checks and reversible migration in ResiCache; verified failure semantics with real Redis/Cluster and concurrency tests.
- **Open-source collaboration**: Two merged codebase-memory-mcp contributions cover client integration, Linux installer compatibility, installation regressions and user-content preservation.

Internship Experience

Beyondsoft

2026.01 - 2026.06

Backend Developer Intern | Alibaba Cloud Private Cloud ·Aegis Host Security

- **Resource integration**: Contributed to Alibaba Cloud private-cloud Aegis security backend development; coordinated with platform/ECS/BMS/BMCP teams and independently built cluster GPU ingestion and aggregation for statistics, filtering and related queries.
- **AI asset identification**: Contributed to the feature and independently built OpenClaw/Ollama test containers; combined container state, GPU relationships and vulnerability data for asset-state assessment, associations and risk displays.
- **Query optimization and fixes**: Adjusted SQL plans, indexes, predicates and covered fields to reduce list/statistics query latency; used Arthas `watch` / `trace` / `stack` for call paths, timings, exceptions and parameters, independently fixing defects and verifying regressions.
- **Image trust-chain diagnosis**: Located containerd TLS failures pulling Harbor private images; checked CA, `certs.d`, `hosts.toml`, `imagePullSecrets` and node caches, then documented diagnostic scripts and checklists.

Project Experience

UltiCode Online Programming Judge GitHub

2025.10 - Present

Tech stack: Java 17 / Spring Boot 4 / Spring Security / MyBatis-Plus / MySQL / Redis Streams / Dubbo / Flyway / Docker

Personal open-source judge covering practice, contests, submissions, sandbox judging and administration. Submission owns submission data; a separate Judge Worker handles delivery failures, duplicates and rejudge-result races.

- **Reliable delivery and recovery**: Commits submissions and Outbox records together; updates delivery state with backoff retries. Lua atomically combines deduplication and Redis Streams enqueueing to avoid mark-only failures. `XPENDING` / `XCLAIM` reclaim timed-out unacknowledged work; capacity/retry limits and dead letters bound invalid redelivery.
- **Rejudge-result race control**: Uses `generation` + `attemptId` to identify valid execution, with leases and heartbeat renewal. Verdict SQL CAS checks both conditions to prevent expired attempts overwriting a newer judging round.
- **Untrusted-code constraints**: Uses Docker network isolation, non-root execution, a read-only root filesystem, capability removal, and Seccomp, plus CPU, memory, PID, and file-descriptor limits to constrain code permissions and resources.
- **Refresh-token replay protection**: Stores only SHA-256 refresh-token hashes. Rotation checks purpose, ownership, and expiry; conditional updates consume an old token only once and issue its replacement in the same transaction to handle concurrent refreshes.
- **CI and release verification**: Added real-MySQL fresh/upgrade migration and browser/Judge acceptance to CI (login, AC/WA, rejudging, rankings). Verifies signatures, SBOMs, SLSA provenance and tag digests across nine images; rejects incomplete/conflicting artifacts. Acceptance is separate from production deployment.

ResiCache Cache Protection Middleware GitHub

2025.01 - Present

Tech stack (current development line): Java 21 / Spring Boot 4 / Spring Cache / Redis / Redisson / Micrometer /

Testcontainers

Personal open-source Spring Cache extension combining enhanced annotations and a handler chain for same-key request coalescing, distributed loading coordination, versioned early expiration and secure deserialization.

- **Same-key concurrency and async deduplication**: `ConcurrentHashMap` + `CompletableFuture` coalesce JVM requests. A distributed-lock Leader double-checks and loads; Followers share results. Per-key in-flight Futures, bounded queues and `CallerRunsPolicy` control backlog and completion/registration cleanup races.
- **Layered Bloom filtering**: Local hits skip remote checks; misses query Redis and backfill accordingly, without rejecting valid loads solely on a local miss.
- **Versioned early expiration**: Redis Lua atomically checks cache-value version before shortening TTL, so stale async tasks cannot alter a new value's lifetime. Later requests load after expiry, handling races with updates and deletion.
- **Deserialization protection**: Jackson streaming validates the type allowlist before parsing versioned envelopes. Regressions cover malicious types and metadata round trips without building a full JSON tree during validation.
- **Failure semantics and validation**: Separates business-load exceptions from cache-write failures, returning a successfully loaded value even if write-back fails. Concurrent and real-Redis integration tests cover shared results, exception propagation, new-value TTL protection, and no resurrection after deletion.
- **Cluster and migration safety**: Verified cache/lock-key slot co-location on a real three-master Redis Cluster. Serialization migration supports shadow reads, dual writes, cutover, rollback, TTL preservation, CAS protection against concurrent-value overwrites and partial-failure resumption.

Professional Skills

Java backend and data processing

- Familiar with Java, Spring Boot, Spring MVC, Spring Security, and MyBatis-Plus, with API design, authentication/authorization, and exception-handling experience.
- Familiar with MySQL index design, execution-plan analysis, and slow-SQL tuning; experienced with Redis cache protection, distributed locks, and concurrent loading.
- Project experience with Dubbo, Redis Streams, and Outbox; understands transaction boundaries, duplicate consumption, retries, and eventual consistency.

Diagnosis and engineering validation

- Combines Arthas, Linux process information, container logs, and SQL plans to diagnose issues, implement fixes, and verify regressions.
- Familiar with Docker images, networking, resource limits, and basic isolation, with containerd/Harbor/private-image TLS troubleshooting experience.
- Familiar with Git, Maven, Docker Compose, Flyway, and Micrometer; uses Testcontainers and concurrent tests for failure/race scenarios.

Education & Honors

Education

Luoyang Normal University

Data Science and Big Data Technology ·Bachelor's degree

GPA: 3.6 / 4.0 (2021.09 - 2026.07)

Honors

- 2025 Lanqiao Cup Java Software Design Group B ·**National Third Prize**
- 2024 Teddy Cup Data Analysis Skills Competition ·**National Third Prize**
- 2023 National University Computer Skills Challenge ·**National Third Prize**

Open-source Contributions and Technical Writing

Open-source contributions

codebase-memory-mcp: Contributed Oh My Pi client integration, MCP registration, install/uninstall and user configuration protection; fixed Linux trusted system-path compatibility and added installer regression coverage.

Selected technical knowledge pages

- Java production performance diagnosis: a minimal symptom-to-evidence decision tree: Connects process/thread stacks, GC, and Arthas observations into containment, evidence preservation, diagnosis, and recovery.
- UltiCode Outbox and Redis Streams: Making Judge Delivery Recoverable: Explains transactional outbox, atomic delivery, retries, dead letters and pending-message recovery.