Helian Peng (David)



Java Backend / Agent Application Developer
2026 graduate combining Java backend development with Python/LangGraph Agents, focused on business integration, identity isolation, source boundaries and explicitly confirmed writes.
Chongqing
Recruiters interested in my experience are welcome to contact me by email. ·Email · lysf15520112973@163.com ·GitHub ·DavidHLP ·Knowledge base

Selected Strengths

- **Java APIs + Agents**: Built controlled tool calls, sourced analysis and multi-turn conversations with Python/FastAPI and LangGraph over existing Java identity and data authorization.
- **Recoverable operations**: Used owner checks, draft versions, explicit confirmation and same-key reconciliation for cancellation, stale confirmation and unknown write results; separates implementation from formal acceptance.
- **Boundary verification**: Used two-account HTTP controls, real-model adversarial isolation and citation checks, covering identity replacement, cross-account requests and source injection.

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

Java 17 / Spring Boot 4 / Spring Security / MySQL / Redis Streams / Dubbo / Flyway / Docker

Personal open-source judge: Java services own identity, problems, submissions and business records; an independent Judge Worker executes judging. The Agent integrates through existing HTTP contracts.

- **Reliable task delivery**: Writes submission and Outbox records in one transaction, delivers to Redis Streams with backoff, and combines deduplication marking with enqueueing in Lua. Timeout recovery and dead letters handle failure windows.
- **Execution-result races**: Uses `generation` + `attemptId`, leases and heartbeats to identify valid attempts. SQL CAS rejects stale verdicts; Docker constrains untrusted-code permissions and resources.
- **Identity and acceptance**: Hashes refresh tokens, consumes each once and rotates within a transaction. Real-MySQL migrations and browser/Judge login, AC/WA, rejudge and ranking acceptance run in CI.

UltiCode Agent Business Integration (Experimental) Source

Python / FastAPI / httpx / LangGraph / SQLite / SSE / MCP / Java Auth & App API

Controlled analysis, multi-turn conversations and optional learning plans for the owner's ordinary practice submissions. Java retains identity and business-data authority; the Agent manages private conversations and local workflow state.

- **Trusted identity and tool boundaries**: Resolves identity through Java `/auth/me` with an isolated cookie client per request. Exposes only allowlisted read-only tools, validates arguments and projects returned fields; the model cannot choose or rewrite ownership.
- **Bounded model/tool loop**: Orchestrates model, tool and result steps with LangGraph, limiting rounds and total duration. Separates tool failure, limit exhaustion and cancellation; tool results remain data, with no authority to issue privileged instructions.
- **Conversations and checkpoints**: Tracks owner, conversation/turn, thread/run and draft version in SQLite. Recovery rechecks ownership; cancellation fences prevent stale runs saving results. Conversation history and learning-plan threads remain distinct.

- **Explicit confirmation and reconciliation**: Editing invalidates old confirmation; saving requires separate confirmation bound to the current version and digest. Unknown Java write outcomes remain unresolved and are reconciled by same-key readback, never blind retries with a new key.
- **Sources and MCP projection**: Preserves source scope, version and position. Authorized source code stays request-scoped, outside conversations, events and checkpoints. Optional stdio MCP exposes only the owner's submission list through the authenticated client and strict field projection.

Reviewed stage-acceptance records for two-account HTTP/model isolation and sourced analysis. LearningPlan writes and failure recovery are not claimed as formally accepted. Live analysis and persistence require separate opt-ins; no production deployment or real-user pilot is claimed.

Professional Skills

Java backend and reliability

- Familiar with Java, Spring Boot, Spring Security and MyBatis-Plus; practical API design, authentication/authorization, MySQL tuning and transaction boundaries.
- Project experience with Redis cache protection, distributed locks, Streams, Outbox and Docker isolation; understands retries, duplicate consumption and eventual consistency.
- Uses Arthas, SQL plans, Linux and container logs for diagnosis, with concurrent tests, Testcontainers and CI to verify failure scenarios.

Agent applications and business integration

- Project experience with Python, FastAPI, httpx and LangGraph, integrating existing Java APIs with tool calls, timeouts, cancellation and streamed events.
- Practices server-trusted identity, allowlisted tools, field projection, source consent and context boundaries; separates model suggestions, user confirmation and business writes.
- Practices SQLite conversations/checkpoints, ownership/version checks, idempotent saves and unknown-outcome reconciliation; distinguishes controlled tests, real-service validation and production delivery.

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.