Ray Tse

AI Agent Engineer · Backend Engineer · Age 18

Seeking a full-time AI Agent Engineer / Backend Engineer role · Remote or hybrid · Available immediately

Software engineer across AI agents and backend systems. I turn real business requirements into working agents — task breakdown, tools, permission boundaries — and own their runtime, context, memory, evals and observability.
Backend work is mainly Java (Spring Boot, Spring Cloud microservices), plus TypeScript / Node.js services, from API to deployment.

ray@raytse.io +86 14716114439 github.com/TsezzzRay raytse.io

EXPERIENCE

Backend Engineer · HQNEAT — OathXPay

2025.11 – 2026.03

Java 17 · Spring Boot 3 · Spring Cloud Alibaba · Nacos · OpenFeign · MyBatis-Plus · MySQL · Redis — cross-border payments microservice platform; owner of the operations console service.

- Delivered the operations console microservice from zero (Spring Boot 3, Spring Cloud Alibaba, Nacos): 80+ REST endpoints across 10+ business modules — merchants, KYB, risk, order review, reconciliation.
- Connected 5 downstream services (user, account, order, withdrawal, config) through OpenFeign, relaying login state with a RequestInterceptor for unified cross-service auth.
- Shipped Redis-backed distributed sessions and Spring AOP data scoping with button-level RBAC, so sales staff see only their own merchants.
- Built the MyBatis-Plus data layer (MySQL schema, shared pagination and responses, audit-field auto-fill) and the EasyExcel reconciliation export; fixed boot failures from Spring Boot 3 auto-configuration changes and a bean circular dependency.

PROJECTS

OpenScreen `github.com/TsezzzRay/openscreen`

2026.07 – now

Open source · macOS desktop agent · TypeScript, Node.js, Electron, React, Swift

- Built solo a macOS Computer Use agent: an agent loop with multi-step Function Calling that reads the UI from multimodal screenshots and the Accessibility tree and acts via file, shell and desktop tools.
- Context engineering: automatic and manual Context Compaction against the model's context window, with a provenance rule that keeps summaries from turning tool output into user goals; token-budgeted long-term memory and screen-activity memory injected each turn and traceable to source frames.
- Safety and human-in-the-loop: Seatbelt-sandboxed Bash; least-privilege approval for each file change, host command and desktop action; memory and tool output injected as untrusted data against Prompt Injection; cancellation voids pending approvals and flags uncertain side effects.
- Evals and observability: five agent eval suites (tasks, file / Bash security, desktop security, memory, compaction) with deterministic checks plus LLM-as-Judge grading (latest 32/34, 5/5, 8/8, 9/9, 12/12); turn-level tracing of inference, tokens, tool calls and approvals separates model, tool and runtime failures.

SKILLS

Tech stack

AI / Agent	LangChain, MCP, Function Calling, RAG, Prompt / Context Engineering, Agent Memory, Multi-Agent, Computer Use, LLM-as-Judge evals
Java	Java, Spring Boot, Spring MVC, Spring Cloud Alibaba (Nacos, OpenFeign), MyBatis-Plus, Maven
TS/Python	TypeScript, Node.js, Python, FastAPI, REST API, WebSocket
Data	MySQL, Redis, PostgreSQL, MongoDB, SQLite
Tools	Claude Code, Codex, Docker, Linux, Git, Nginx / Caddy

Skills overview

- Proficient in Java 8+, OOP and design patterns; Spring Boot, Spring MVC and Spring Cloud microservices — service discovery, remote calls, layered architecture.
- Data modeling, persistence and session management with MySQL, Redis, MongoDB and MyBatis-Plus.
- TypeScript / Node.js server-side development and inter-process communication; REST services in Python with FastAPI.
- Agent frameworks such as LangChain, MCP, Function Calling and RAG; designs an agent's tools, context, memory, permissions and evals from real business requirements.
- Fluent with frontier AI coding tools — Claude Code and Codex — for reading large codebases, implementing features, running tests and tracking down regressions.
- Docker, Linux, Git, Maven and Postman; independent development, system design and debugging.