



Building Local Agentic AI with Pi

- 1 Day Course
- Hands-on and Lecture

Course Overview

This one-day hands-on course provides a practical exploration of modern Agentic AI by opening up the underlying machinery that turns an LLM into an acting system.

Participants begin by establishing clearly that an LLM alone is not an agent. Students analyze major framework paradigms in the industry (LangGraph, CrewAI, AutoGen, and application-owned loops) to understand what those abstractions hide. They then strip away those abstractions and build a minimal, application-owned agent harness from scratch in vanilla JavaScript, using Pi, a small, real, MIT-licensed agent harness, running entirely locally against Ollama and a small local model, with no cloud API keys and no licensing headaches.

By the end of the day, participants will have built a functioning application-owned AI agent directly against a local LLM endpoint, including tool calling, execution dispatch, observation handling, message history/state, and loop termination, culminating in a capstone diagnostic agent that investigates a broken web application and recovers from a real tool failure on its own.

Who Should Attend

- Software Engineers
- Systems Developers
- AI/ML Solutions Architects
- Technical Leads

What You'll Learn

- Define the LLM/Harness Boundary: Differentiate clearly between model generation capabilities and the external control structures required to turn an LLM into an acting system.
- Deconstruct the Agent Framework Landscape: Evaluate the design trade-offs and abstractions of explicit state graphs (LangGraph), role-oriented teams (CrewAI), conversational networks (AutoGen), and application-owned loops, and understand why Pi, a minimal application-owned harness, is the clearest vehicle for learning the control loop directly.
- Build an Application-Owned Agent Harness: Implement a complete, working agentic loop from scratch in vanilla JavaScript directly against a local LLM endpoint.
- Implement Core Agent Control Mechanics: Apply tool calling, execution dispatch, observation handling, message history/state, and loop termination in a functioning agent.

Outline

1. Foundations: What Makes Something an Agent

- Lecture: What Makes Something an “Agent”?, covering why an LLM alone has no memory and no authority to act, and the core agentic loop: State → LLM → Tool Call → Dispatcher → Execution → Observation → Updated State → Repeat/Stop
- Hands-on Lab: Setting Up Your Agent Environment, installing Node.js, Ollama, a local model, and the Pi agent harness, then running and modifying a working reference agent

2. Tool Contracts, Instructions, and Failure

- Lecture: Tool Contracts, Instructions, and Failure, covering how a model learns a tool exists, why instructions shape format but not truth, and what can go wrong along the way
- Hands-on Lab: Turning a Function into a Tool the Model Can Call, writing a tool description and manually carrying out the full tool-call round trip
- Hands-on Lab: Giving the Agent a Toolbox, registering multiple tools and watching the model choose the correct one on its own
- Hands-on Lab: Running Multiple Tools in One Turn, handling a single model turn that requests more than one tool at once
- Hands-on Lab: Handling a Tool Call Gone Wrong, catching a malformed tool call with schema validation before it crashes the program
- Hands-on Lab: Steering the Model with Instructions, reshaping model behavior with only a system prompt and seeing the limits of that lever

3. Building the Agentic Loop

- Lecture: The Agent Framework Landscape, comparing LangGraph, CrewAI, AutoGen, and application-owned loops, and explaining why this course builds the loop by hand with Pi
- Hands-on Lab: Building the Repeating Loop, automating the manual round trips from earlier labs into a loop that calls the model until it has a real answer
- Hands-on Lab: Teaching the Agent When to Stop, adding the real stop condition a loop needs instead of an artificial round limit
- Hands-on Lab: The Diagnostic Agent (Capstone), combining every mechanic from the day into one agent that investigates a real problem and recovers from a genuine tool failure

4. Beyond Today

- Lecture: Beyond Today: Production-Grade Agents, previewing what a production agent needs beyond a single day: surviving a crash, human-in-the-loop authorization, sandboxed execution, and multi-agent orchestration

Prerequisites

- Basic familiarity with JavaScript/TypeScript syntax (functions, loops, objects). No prior professional JavaScript experience required: all exercises are guided, with working code provided at every step.
- Familiarity with REST APIs, HTTP protocols, and JSON parsing
- Basic working knowledge of Linux command-line environments

Next Courses

- Advanced follow-on course covering production hardening topics: durable state persistence, checkpoint recovery, Human-in-the-Loop (HITL) authorization, sandboxed execution, and multi-agent orchestration