Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Building Custom Tools and Skills for AI Coding Agents

Class Duration

7 hours of live training delivered over 1-2 days to accommodate your scheduling needs.

Student Prerequisites

- Professional software development experience in TypeScript or Python
- Familiarity with REST APIs and JSON Schema

Target Audience

Software engineers and platform engineers who want to extend AI coding agents - Claude Code, GitHub Copilot via MCP, or LLM-powered internal tools - with custom capabilities, safe execution sandboxes, and tool pipelines tailored to their organization's systems. Teams packaging tools as servers can continue with *Model Context Protocol (MCP) for Developers*, and those wiring tools into autonomous agents can move on to *Building AI Agents with Python and MCP*.

Description

This course covers the full lifecycle of building reliable custom tools for AI agents: from tool design and schema authoring through safe execution, error handling, and multi-tool composition. We treat tool-building as software engineering - with correctness, security, and maintainability as primary concerns - not just as LLM plumbing. Topics include function/tool calling mechanics across major model APIs, designing tool interfaces that models use reliably, safe execution sandboxes for code-running tools, and composition patterns for multi-step tool pipelines.

Learning Outcomes

- Describe how function/tool calling works in Claude, OpenAI, and Gemini APIs.
- Design tool schemas (name, description, parameter JSON Schema) that models invoke accurately.

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- Implement tool handlers with proper validation, error propagation, and structured responses.
- Build safe execution sandboxes for tools that run arbitrary code or execute system commands.
- Compose multi-tool pipelines with appropriate sequencing and state passing.
- Apply security best practices: sandboxing, input sanitization, capability scoping, and audit logging.
- Package and distribute custom tools as MCP servers or agent plugin modules.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Python 3.12+ or Node.js 22+, API access to at least one frontier model, and Docker (for sandbox labs).

Training Topics

Function/Tool Calling Mechanics

- How tool calling works at the API level
- Tool call request and result round-trip
- Parallel tool calls and sequential dependencies
- Differences across Claude, OpenAI, and Gemini APIs

Tool Schema Design

- Name and description as model-facing interface
- JSON Schema for parameters: types, enums, required fields
- Writing descriptions that models use correctly
- Anti-patterns: over-broad schemas, ambiguous names

Tool Handler Implementation

- Validation and error propagation patterns
- Structured success and error responses
- Idempotency and retry safety
- Logging and observability for tool invocations

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Safe Execution Sandboxes

- Risk categories: file system, network, process execution
- Docker-based isolation for code-running tools
- Resource limits: CPU, memory, and timeout enforcement
- Capability scoping: minimal permission principle

Multi-Tool Composition

- Sequential pipelines and data passing
- Conditional branching based on tool results
- Aggregating results from parallel tool calls
- Tool pipeline debugging and tracing

Security and Governance

- Input sanitization for command-execution tools
- Preventing prompt injection through tool outputs
- Audit logging: who invoked what, when, with what args
- Tool approval workflows for sensitive capabilities

Packaging and Distribution

- MCP server packaging for Claude and Copilot
- Plugin modules for internal agent frameworks
- Versioning and compatibility management

