Multi-Agent Systems with CrewAI

- 2 Days Course
- Hands-on

Course Overview

This 2-day hands-on workshop teaches engineers how to design, orchestrate, and deploy collaborative multi-agent teams using CrewAI. Participants learn how to move beyond single-prompt setups by structuring specialized roles, managing sequential and event-driven workflows with CrewAI Flows, integrating custom tools, and implementing enterprise guardrails for production-grade reliability.

Who Should Attend

- Software Engineers
- Tech Leads
- AI Application Developers

What You'll Learn

- Transition from monolithic prompting to modular, role-based agent architectures using CrewAI primitives.
- Build, ground, and safely deploy custom tools using BaseTool and type-hinted decorators.
- Implement deterministic, event-driven workflows using CrewAI Flows and manage persistent state.
- Apply enterprise guardrails, human-in-the-loop checkpoints, and open-source observability tracing.
- Orchestrate an end-to-end autonomous research flow in a hands-on capstone project.

Outline

1. Foundations of Role-Based Multi-Agent Systems

- From Single Prompts to Agent Teams: Transitioning from monolithic prompting to modular, role-based agent architectures.
- Core Mental Model: Deconstructing the CrewAI primitives: Agents, Tasks, Tools, and Crews.
- Defining Personas & Model Selection: Configuring role, goal, and backstory; matching task complexity to frontier vs. lighter models.
- Task Contracts & Structured Outputs: Enforcing explicit expected outcomes and output schemas using Pydantic for reliable downstream parsing.

2. Custom Tools, Environment & Safety

- Rapid Environment Setup: Utilizing pre-configured .env boilerplate and project scaffolding to manage credentials securely without setup friction.
- Building Custom Tools: Subclassing BaseTool and applying type-hinted decorators to expose local databases, web scrapers, and internal APIs.
- Tool Grounding & Exception Handling: Preventing hallucinated parameters, handling runtime API errors gracefully, and setting tool limits to eliminate infinite retry loops.

3. Advanced Orchestration, Flows & Memory

- Orchestration Paradigms: Knowing when to use Hierarchical Managers (LLM-driven delegation) versus CrewAI Flows (deterministic, event-driven control).
- Event-Driven Workflows with CrewAI Flows: Structuring production logic using @start(), @listen(), and @router() decorators for dynamic branching.
- State & Memory Management: Managing persistent flow state, passing structured Pydantic payloads across execution boundaries, and configuring entity memory.

4. Enterprise Guardrails, Observability & Capstone

- Production Guardrails: Setting iteration limits, rate-limiting LLM calls, and implementing Human-in-the-Loop (HITL) approval steps for sensitive task execution.
- Observability & Tracing: Instrumenting open-source tracing (Langfuse / Phoenix) for token usage, latency analysis, and execution debugging.
- 3-Hour Capstone Project: Complete an Autonomous Research & Intelligence Flow. Working from a skeletal framework, engineers write custom tool logic, configure task validation contracts, implement human-in-the-loop validation, and emit a structured, multi-source research report.

Prerequisites

- Proficiency in Python
- Basic familiarity with REST APIs
- Understanding of basic LLM prompting