Faysal Aziz

Agentic AI in Action

Building Autonomous Agents in 2 days

FA-0049 · 2 days

Agentic & Generative AI · Software Development

Introduction

From Code to Autonomy: Empowering Developers to Build Intelligent Agents for Real-World Applications

AI is changing fast, and one of the most exciting developments is the rise of "agentic AI" – basically, AI that doesn't just sit there waiting for commands. Instead of the old question-and-answer style we're used to, these new AI systems can actually look around, figure things out, and take action on their own to get stuff done. Think of tools like OpenAI's Operator and Manus – they're already showing us what this looks like in the real world, handling web tasks and fitting right into our daily workflows without much hand-holding.

If you're a Python developer who knows their way around code, this two-day deep dive is for you. We're not just talking theory here – you'll actually build, launch, and manage your own AI agents. You'll work with APIs, get hands-on with platforms like Operator and Manus, and most importantly, learn how to create solutions that actually solve real problems. By the end, you'll have the skills to turn these powerful concepts into tools that work in the real world.

Learning outcomes

By the end of this course, participants will be able to:

- **Understand** the fundamental principles and architecture of agentic AI systems.
- **Differentiate** between traditional AI models and agentic AI in terms of autonomy, adaptability, and decision-making capabilities.

- **Develop** AI agents using Python, incorporating APIs for enhanced functionality.
- **Integrate** and utilize OpenAI's Operator and Manus platforms for deploying agentic AI solutions.
- **Implement** best practices for building secure, efficient, and scalable AI agents.
- **Evaluate** the ethical considerations and potential risks associated with deploying autonomous AI agents in various domains.

Prerequisites

Participants should have:

- A strong command of Python programming.
- Familiarity with RESTful APIs and web development concepts.
- Basic understanding of machine learning principles.
- Experience with version control systems like Git.
- Access to a development environment with Python 3.8+ installed.

Training outline

1. 1. Foundations of Agentic AI

1.1. Introduction to Agentic AI

- Evolution from traditional AI to agentic AI.
- Key characteristics: autonomy, adaptability, and goal-oriented behavior.
- Real-world applications and case studies.

1.2. Architectural Overview

- Core components: perception, reasoning, and action modules.
- Communication protocols and data flow within agentic systems.
- Comparison with traditional AI architectures.

1.3. Ethical and Societal Implications

- Autonomy vs. control: balancing agent independence with human oversight.
- Addressing biases and ensuring fairness in decision-making.

- Regulatory frameworks and compliance considerations.

2. 2. Building AI Agents with Python

2.1. Setting Up the Development Environment

- Required libraries and tools: requests, aiohttp, FastAPI, etc.
- Environment configuration and best practices.

2.2. Designing the Agent's Core

- Defining agent objectives and success criteria.
- Implementing perception modules: data collection and preprocessing.
- Developing reasoning engines: decision-making algorithms and logic.

2.3. Action Modules and Execution

- Integrating APIs for task execution.
- Handling asynchronous operations and concurrency.
- Error handling and recovery mechanisms.

2.4. Persistence and Learning

- Implementing memory systems for state retention.
- Incorporating learning mechanisms for adaptability.
- Data storage solutions and management.

3. 3. Integrating with OpenAI's Operator

3.1. Overview of OpenAI's Operator

- Capabilities and use cases.
- Understanding the o3 model and its enhancements.

3.2. Authentication and API Access

- Setting up API keys and managing access.
- Navigating OpenAI's API documentation.

3.3. Implementing Operator in Python

- Sending tasks and interpreting responses.
- Managing sessions and maintaining context.

3.4. Advanced Features and Customization

- Utilizing Operator's browsing and interaction capabilities.
- Customizing behavior through prompts and configurations.

4. 4. Leveraging Manus for Agentic AI

4.1. Introduction to Manus

- Understanding Manus's architecture and functionalities.
- Use cases in various industries.

4.2. Accessing Manus Services

- Setting up accounts and managing credentials.
- Exploring available APIs and endpoints.

4.3. Building Agents with Manus

- Creating tasks and workflows.
- Monitoring and analyzing agent performance.

4.4. Comparative Analysis

- Evaluating Manus vs. Operator: strengths and limitations.
- Selecting the appropriate platform based on project requirements.

5. 5. Advanced Agent Development

5.1. Multi-Agent Systems

- Designing systems with multiple interacting agents.
- Communication protocols and coordination strategies.

5.2. Security and Compliance

- Implementing authentication and authorization mechanisms.
- Ensuring data privacy and protection.

5.3. Deployment and Scaling

- Containerization with Docker.
- Deploying agents on cloud platforms.
- Monitoring and logging for maintenance.

5.4. Testing and Validation

- Unit and integration testing strategies.
- Performance benchmarking and optimization.

6. 6. Capstone Project

6.1. Project Planning

- Defining objectives and success metrics.
- Allocating resources and setting timelines.

6.2. Development Phase

- Building a functional agent using Python and integrating with Operator or Manus.
- Implementing features learned throughout the course.

6.3. Presentation and Review

- Demonstrating the agent's capabilities.
- Peer review and feedback sessions.

6.4. Reflection and Next Steps

- Discussing challenges faced and solutions implemented.
- Exploring opportunities for further development and learning.

Instructor Profile:

This course is led by an industry veteran with over 30 years of experience in software development and artificial intelligence. The instructor brings a wealth of practical knowledge, having worked on numerous real-world projects that leverage agentic AI. Participants will benefit from insights into industry best practices, common pitfalls, and emerging trends, ensuring a learning experience grounded in practical application rather than theoretical concepts.

By the end of this intensive two-day course, participants will be equipped with the skills and knowledge to design, develop, and deploy agentic AI systems using Python, OpenAI's Operator, and Manus. This course serves as a stepping stone for developers aiming to harness the power of autonomous agents in solving complex, real-world problems.

© 2026 Faysal Aziz · Kuala Lumpur