

SAMIR GENAI ACADEMY

PROFESSIONAL STUDY GUIDE SERIES

Phase 03 · Agentic AI Engineering

12

OpenAI Agents SDK

Build production agents with Swarm, handoffs & guardrails

Module 12 of 25 · 1 hour

Instructor: **Samir Kumar** · Edition 2026

STUDY GUIDE

01 Overview

Build production agents with Swarm, handoffs & guardrails. This guide gives you the core theory, the tools that matter, a hands-on build, working code, and the pitfalls to avoid — everything you need to learn the topic to a professional standard and apply it in real projects.

1 hour

LESSON LENGTH

4 + 4

CONCEPTS + BUILD
STEPS

3

FRAMEWORKS & TOOLS

12/25

PHASE 03

02 Learning Objectives

- ▶ Use the Agents SDK to define agents, tools, and instructions.
- ▶ Implement handoffs between specialised agents.
- ▶ Add guardrails for safety and input/output validation.
- ▶ Ship a multi-agent workflow.

03 Core Concepts

Agents SDK model

The SDK formalises an agent as instructions + tools + model, with a runtime that manages the tool-calling loop, tracing, and sessions. It removes boilerplate so you focus on behaviour and orchestration.

Handoffs

A handoff lets one agent delegate to another specialised agent (triage -> billing -> refunds). Handoffs keep each agent's instructions focused and let you compose teams that route work to the right expert.

Guardrails

Guardrails validate inputs and outputs — block off-topic requests, enforce output schemas, and run safety checks in parallel with the agent. They are essential for production trust and compliance.

Tracing & sessions

Built-in tracing records each step for debugging and evaluation; sessions persist conversation state. Together they make multi-agent systems observable and maintainable.

04 Visual Diagram



Figure 12 — A triage agent routes each request to the right specialist; guardrails validate the result.

05 Key Frameworks & Tools

[OpenAI Agents SDK](#)[Pydantic \(guardrail schemas\)](#)[Tracing dashboard](#)

06 Hands-On Build

🔗 SDR Sales Agent with Multi-Agent Handoffs

1. Define a triage agent plus specialist agents (qualify, draft, schedule).
2. Give each agent focused instructions and the tools it needs.
3. Wire handoffs so triage routes leads to the right specialist.
4. Add input/output guardrails and review the trace.

07 Code Walkthrough

Defining agents and a handoff

PYTHON

```
from agents import Agent, Runner

refunds = Agent(name="Refunds", instructions="Handle refund requests.")
billing = Agent(name="Billing", instructions="Answer billing questions.")
triage = Agent(
    name="Triage",
    instructions="Route the customer to the right specialist.",
    handoffs=[refunds, billing],
)
result = Runner.run_sync(triage, "I was double charged last month.")
print(result.final_output)
```

08 Lab Assistant

Lab 12 • Multi-agent handoff

Prerequisites: pip install openai-agents • An API key

1. Define a triage agent with handoffs to two specialists

Expected: Runner routes the query to the correct agent

2. Add an output guardrail (schema/safety)

Expected: Invalid outputs are blocked

3. Inspect the run trace

Expected: Each step (route, tool, output) is visible

Troubleshooting

- Wrong routing → clarify each agent's instructions.
- Guardrail too strict → relax the schema or rules.

09 Pitfalls & Key Takeaways

⚠ COMMON PITFALLS

- One mega-agent with overloaded instructions instead of focused handoffs.
- Skipping guardrails on user-facing agents.
- Not reading traces, so silent routing errors persist.

✓ KEY TAKEAWAYS

- Agents = instructions + tools + model, managed by a runtime.
- Handoffs compose focused specialists into a team.
- Guardrails + tracing make agents production-safe and debuggable.

 **Companion video:** Watch the module 12 lesson in your Student Dashboard, then use this guide to revise, copy the code, and complete the build. Download both for offline study.

10 Further Resources

OpenAI Agents SDK docs • Swarm patterns • Guardrails examples