# Build Your First AI Agent in Python — Resource Guide

Everything you need to follow along with the video, in the same order. Agentspan is free and open source.

## Prerequisites

Install these three before you start:

1. **Python 3.10 or newer** — https://www.python.org/downloads/ (click the yellow download button at the top).
2. **Java Development Kit (JDK)** — https://adoptium.net/temurin/releases. Pick your operating system and install the latest long-term support (LTS) release. JDK 21 also works fine if you need to stay on it.
3. **uv (Python package manager)** — https://docs.astral.sh/uv/getting-started/installation/

Install uv:

On Windows, refresh your PATH so `uv` is available without reopening the terminal:

```
None
$env:Path = "$HOME\.local\bin;$env:Path"
```

## Create the Project

In VS Code, use **File ➜ Open Folder** to open an empty folder, then open the terminal with **View ➜ Terminal**.

```
None
uv init
uv add agentspan
```

Sanity-check the install:

```
None
uv run agentspan doctor
```

Green output means you're ready. If you see red, it's almost always one of the two issues below.

## Set Up Your AI Provider and API Key

This video uses OpenAI. Anthropic and a local Ollama model both work too.

- **OpenAI:** https://platform.openai.com/api-keys ➜ Create new key
- **Anthropic:** https://platform.claude.com/ ➜ Manage API keys ➜ Create key

Set the matching environment variable:

```
None
# Mac / Linux
export OPENAI_API_KEY=sk-...

# Windows (PowerShell)
$env:OPENAI_API_KEY="sk-..."
```

(Use `ANTHROPIC_API_KEY` instead if you went with Anthropic.)

**Fix:** **`CERTIFICATE_VERIFY_FAILED`** (common on Mac) — run the certificate installer that ships with Python, matching your installed version:

```
None
# Mac (replace 3.14 with your Python version)
open "/Applications/Python 3.14/Install Certificates.command"

# Windows
pip install --upgrade certifi
```

Re-run `uv run agentspan doctor` and you should be green.

## Your First Agent — `main.py`

Three pieces: a tool, the agent, and one line to run it.

```
Python
from agentspan.agents import Agent, AgentRuntime, tool
```

```python
@tool
def get_weather(city: str) -> str:
    """Get current weather for a city."""
    return f"72 degrees and sunny in {city}"


agent = Agent(
    name="weatherbot",
    model="openai/gpt-5.4-mini",
    tools=[get_weather],
)

with AgentRuntime() as runtime:
    result = runtime.run(agent, "What's the weather in NYC?")
    result.print_result()
```

The model string is always `provider/model-name`. Find OpenAI model names at https://platform.openai.com/docs/models and prefix the name with `openai/`.

Start the Agentspan server in a second terminal (first boot downloads a JAR and takes about 30 seconds; every boot after is instant):

```
None
uv run agentspan server start
```

Run the agent in your first terminal:

```
None
uv run main.py
```

Then open the dashboard to see the execution recorded step by step: **http://localhost:6767**

## Multi-Step Agent — `agent.py`

The agent writes pytest tests for each function in a sample module. Save `price_calculator.py` into your project first.

`price_calculator.py`:

Python

```python
def calculate_discount(price: float, discount_rate: float) -> float:
    """Apply a discount rate to a price. Rate should be between 0 and 1."""
    return price * (1 - discount_rate)


def calculate_tax(subtotal: float, tax_rate: float) -> float:
    """Calculate tax amount on a subtotal. Rate should be between 0 and 1."""
    return subtotal * tax_rate


def calculate_shipping(order_total: float, free_shipping_threshold: float = 50.0)
-> float:
    """Calculate shipping cost. Free shipping on orders at or above the threshold,
otherwise $5.99."""
    if order_total >= free_shipping_threshold:
        return 0.0
    return 5.99


def calculate_total(price: float, quantity: int, discount_rate: float = 0.0,
tax_rate: float = 0.0) -> float:
    """Calculate the final order total including discount, tax, and shipping."""
    subtotal = calculate_discount(price, discount_rate) * quantity
    tax = calculate_tax(subtotal, tax_rate)
    shipping = calculate_shipping(subtotal)
    return subtotal + tax + shipping
```

`agent.py`:

Python

```python
from agentspan.agents import Agent, AgentRuntime, tool


@tool
def save_tests(function_name: str, test_code: str) -> str:
    """Save generated pytest tests for a function to test_price_calculator.py."""
    with open("test_price_calculator.py", "a") as f:
        f.write(f"\n# Tests for {function_name}\n")
        f.write(test_code)
        f.write("\n")
    return f"Tests for {function_name} saved."
```

```
agent = Agent(
    name="testbot",
    model="openai/gpt-5.4-mini",
    tools=[save_tests],
)

# 1. Seed the test file with imports
with open("test_price_calculator.py", "w") as f:
    f.write("import pytest\n")
    f.write("from price_calculator import (\n")
    f.write("    calculate_discount, calculate_tax,\n")
    f.write("    calculate_shipping, calculate_total,\n)\n")

# 2. Read the source file
with open("price_calculator.py") as f:
    source = f.read()

# 3. Run the agent
with AgentRuntime() as runtime:
    result = runtime.run(
        agent,
        f"Write pytest tests for each function below. "
        f"Call save_tests once per function with the function name and test code.
"
        f"test_code must contain only test functions - no import statements. "
        f"Return raw Python only, no markdown.\n\n{source}"
    )
```

Run it (about 15 seconds), then open `test_price_calculator.py` and check the dashboard for the new execution:

```
None
uv run agent.py
```

Optional — prove the generated tests actually pass:

```
None
uv add pytest
uv run pytest test_price_calculator.py
```

## Crash and Resume — `resilient-agent.py`

This version survives a crash. Leave `EXECUTION_ID` empty for a fresh run; paste in an ID to resume a crashed one. Two changes make it work: `save_tests` is **idempotent** (it won't rewrite tests already in the file), and the `time.sleep(2)` gives you a window to kill the process mid-run.

Python

```python
from agentspan.agents import Agent, AgentRuntime, AgentHandle, tool
import time

EXECUTION_ID = ""  # paste an execution ID here to resume a crashed run


@tool
def save_tests(function_name: str, test_code: str) -> str:
    """Save generated pytest tests for a function to test_price_calculator.py."""
    # Idempotency: skip if this function's tests are already saved
    with open("test_price_calculator.py") as f:
        if f"# Tests for {function_name}" in f.read():
            return f"Tests for {function_name} already saved."
    time.sleep(2)  # window to interrupt the run before it finishes
    with open("test_price_calculator.py", "a") as f:
        f.write(f"\n# Tests for {function_name}\n")
        f.write(test_code)
        f.write("\n")
    return f"Tests for {function_name} saved."


agent = Agent(
    name="testbot",
    model="openai/gpt-5.4-mini",
    tools=[save_tests],
)

# Only seed the file on a fresh run, never when resuming
if not EXECUTION_ID:
    with open("test_price_calculator.py", "w") as f:
        f.write("import pytest\n")
        f.write("from price_calculator import (\n")
        f.write("    calculate_discount, calculate_tax,\n")
        f.write("    calculate_shipping, calculate_total,\n)\n")

with open("price_calculator.py") as f:
    source = f.read()
```

Created by David DeWinter | https://linkedin.com/in/daviddewinter

```
prompt = (
    f"Write pytest tests for each function below. "
    f"Call save_tests once per function with the function name and test code. "
    f"test_code must contain only test functions - no import statements. "
    f"Return raw Python only, no markdown.\n\n{source}"
)

if EXECUTION_ID:
    runtime = AgentRuntime.serve()
    handle = AgentHandle(execution_id=EXECUTION_ID, runtime=runtime)
else:
    runtime = AgentRuntime()
    handle = runtime.start(agent, prompt)
    print(f"EXECUTION_ID: {handle.execution_id}")

for event in handle.stream():
    print(event)
```

How to run the crash-and-resume demo:

1. Run `uv run resilient-agent.py` and watch `test_price_calculator.py`.
2. As soon as the first test is written, press **Ctrl+C** to kill the Python process mid-run.
3. Copy the `EXECUTION_ID` printed at the top of the terminal and paste it into the `EXECUTION_ID` variable.
4. Run `uv run resilient-agent.py` again. It resumes the same execution on the server and finishes the remaining tests, with no duplicates.
5. Check the dashboard — same execution ID, all tool calls completed.

## Links

- Agentspan: https://agentspan.ai/
- Agentspan Docs: https://agentspan.ai/docs
- Python: https://www.python.org/downloads/
- Java JDK (Adoptium Temurin): https://adoptium.net/temurin/releases
- uv: https://docs.astral.sh/uv/
- OpenAI API Keys: https://platform.openai.com/api-keys
- OpenAI Models: https://platform.openai.com/docs/models
- Anthropic API Keys: https://platform.claude.com/

# About the Host

David DeWinter is a corporate trainer specializing in AI for business teams. He helps companies move from using it as a chatbot to actually performing repetitive work with it through hands-on workshops, bootcamps, and executive training sessions.

Follow David on LinkedIn. He shares real examples of AI at work in business, with tools and tactics you can use this week.