

OPENAI AGENTS SDK

Custom Agents Quick Reference Guide

Automata Learning Lab | Version 1.0 | December 2024

AT A GLANCE: 4 CUSTOM AGENTS



Research Agent

Web search + notes



Filesystem Agent

Folder cleanup



Dashboard Agent

CSV/Excel analysis



Content Agent

Multi-format output

1. RESEARCH & SUMMARIZATION AGENT

What It Does

- Searches the web for any topic using WebSearchTool
- Returns structured findings: key points, citations, open questions
- Saves research notes to markdown files
- Uses Pydantic ResearchSummary model for typed output

[Learning](#)[Research](#)[Documentation](#)

Usage Example

```
from scripts.custom_agents import run_research_agent

# Research a topic
result = run_research_agent(
    "LLM context windows",
    save_notes=True
)

# Access structured output
print(result.summary)
print(result.key_points)
print(result.citations)
```

2. FILE SYSTEM AUTOMATION AGENT

What It Does

- Scans folders and lists all files with metadata
- Categorizes files by type: Documents, Images, Videos, Code, etc.
- Moves files into organized subfolders
- **Dry-run by default** - shows what would happen first

[Downloads Cleanup](#)[Project Organization](#)

Usage Example

```
from scripts.custom_agents import run_filesystem_agent

# Preview only (safe - no files moved)
result = run_filesystem_agent("~/Downloads")
print(result.summary)

# Actually organize files
result = run_filesystem_agent(
    "~/Downloads",
    execute=True # Set True to move files
)
```


3. PERSONAL DATA DASHBOARD AGENT

What It Does

- Loads CSV or Excel files (expenses, fitness, habits)
- Generates summary statistics (mean, median, min, max)
- Detects anomalies: missing values, outliers, duplicates
- Creates charts: line, bar, scatter, histogram, box plots

Finance Tracking

Fitness Data

Habit Analysis

Usage Example

```
from scripts.custom_agents import run_data_dashboard_agent

result = run_data_dashboard_agent(
    "expenses.csv",
    analysis_focus="spending patterns",
    create_charts=True
)

print(result.key_insights)
print(result.anomalies)
# Chart saved to: ./chart_output.png
```

4. CONTENT PRODUCTION PIPELINE AGENT

What It Does

- Takes transcript, notes, or article as input
- Generates **5 title variations** (curiosity, how-to, listicle)
- Creates **YouTube description** with timestamps & hashtags
- Writes **60-second TikTok/Reels script**
- Produces **newsletter summary** with subject line

YouTube

TikTok/Reels

Newsletter

Usage Example

```
from scripts.custom_agents import run_content_pipeline_agent

transcript = "Today we discuss..."
result = run_content_pipeline_agent(
    transcript,
    content_name="My Tutorial",
    save_to_disk=True
)

print(result.title_variations)
print(result.short_form_script.hook)
# Files saved to: ./content_output/
```

KEY SDK PATTERNS USED

| Pattern | What It Does | Example |
|--------------------------------|--|--|
| <code>@function_tool</code> | Turns a Python function into an agent tool | <pre>@function_tool def my_tool(x: str) -> str:</pre> |
| <code>output_type=Model</code> | Forces structured Pydantic output | <pre>Agent(..., output_type=ResearchSummary)</pre> |
| <code>Runner.run_sync()</code> | Runs agent synchronously | <pre>result = Runner.run_sync(agent, query)</pre> |
| <code>WebSearchTool()</code> | Built-in web search capability | <pre>Agent(..., tools=[WebSearchTool()])</pre> |

**Pro Tip**

All agents return typed Pydantic models. Access fields directly: `result.summary`, `result.key_points[0]`. IDE autocomplete works!

⚠ Safety Note

Filesystem agent uses `dry_run=True` by default. Always preview before setting `execute=True` to avoid accidental file moves.

COMMAND LINE QUICK START

- `python -m scripts.custom_agents.research_agent` - Run research demo
- `python -m scripts.custom_agents.filesystem_agent` - Run filesystem demo
- `python -m scripts.custom_agents.data_dashboard_agent` - Run dashboard demo
- `python -m scripts.custom_agents.content_pipeline_agent` - Run content demo

AUTOMATA LEARNING LAB | OpenAI Agents SDK Course Materials
github.com/openai/openai-agents-python | `pip install openai-agents`