🧠 How the Agent Class Works (Behind the Scenes)

🔹 1. Agent Initialization

The Agent is configured with multiple key components:

- Name: Identifier for the agent.
- Instructions: Guidelines on how the agent should behave.
- Model: The LLM (like GPT) being used.
- Tools: Additional tools the agent can use (e.g., calculators, web).
- Handoffs: Defines if/when tasks are handed to another agent.
- Guardrails: Safety or ethical rules applied.
- Context: User's previous interactions or memory.

🔹 2. User Input

The user provides an input message.

🔹 3. Prompt Creation

The agent combines:

- User Input
- Instructions
- Any context
  ➜ to create a prompt that is sent to the LLM.

🔹 4. LLM Generates Output

The LLM processes the prompt and decides:

- Whether it needs to call a tool
- Or handle the task directly.

🔄 Two Possibilities

✅ A. If Tool Call Is Needed

- The LLM calls a tool.
- That tool is executed.
- The output is returned to the agent.

🔁 B. If No Tool Call

- The system checks if a handoff is required.
  - If yes, the task is:
    - Delegated to a sub-agent.
    - The sub-agent handles it.
    - The result is returned to the main agent.
  - If no handoff, the LLM continues processing.

🔹 5. Final Output

Once the agent (or tool/sub-agent) completes the task:

- It prepares the final output.

🔹 6. Customization & Monitoring

Before sending the final response:

- Apply output formatting.
- Trace & monitor the process (for analytics).
- Apply customization (like temperature, token limits).

🔹 7. Response Returned to User

Finally, the system returns the answer back to the user.

📄 Summary

The Agent Class acts as a smart system that:

- Takes user input,
- Builds a proper query for the model,
- Decides whether to use tools or other agents,
- Processes the task efficiently,

- Monitors and customizes the output,
- And returns a well-formed final response.

Learn With (YT: Subhan Kaladi)