Conversable Agents

Instructor

Lucas Soares

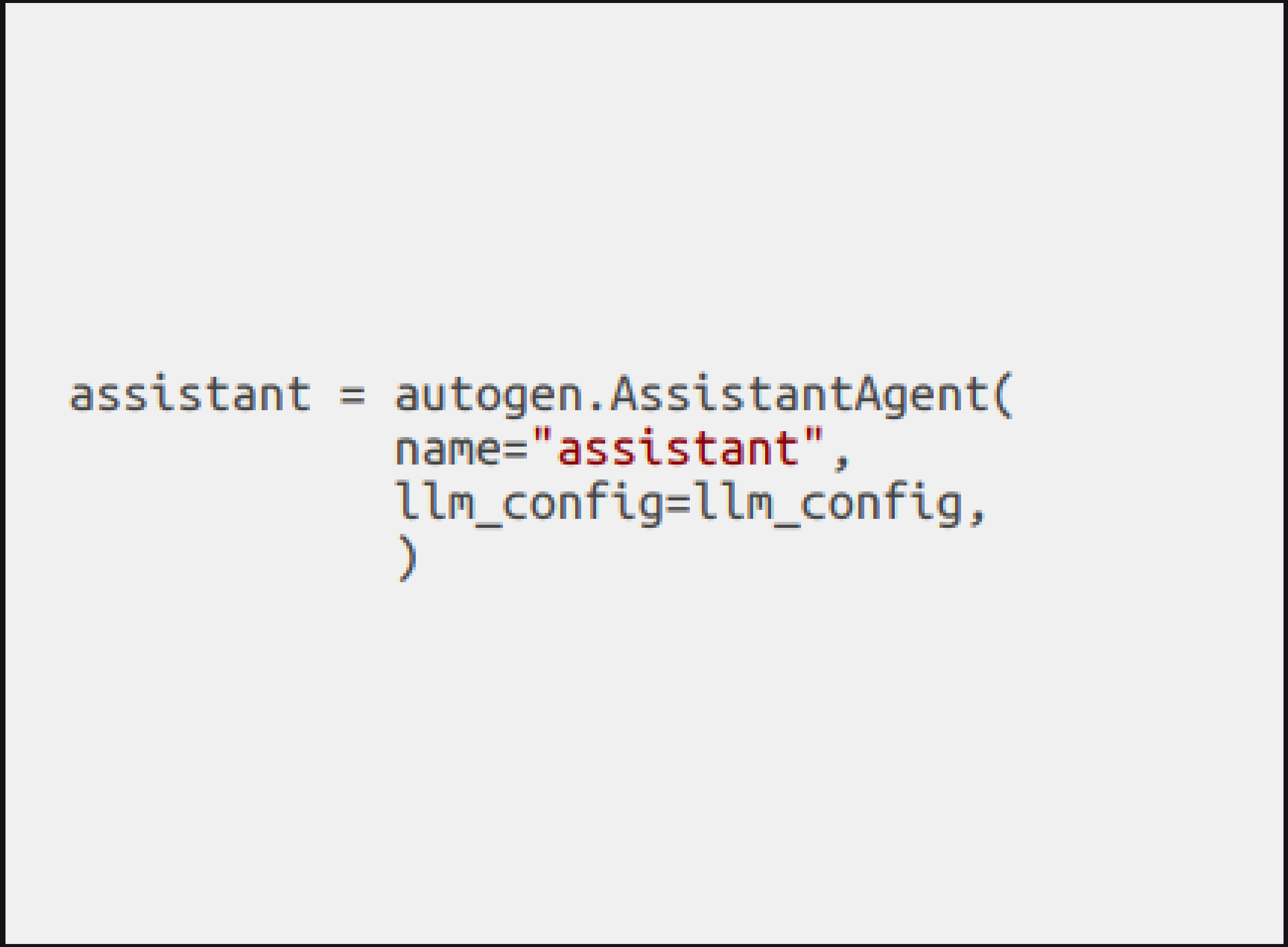
AI Engineer at Otovo
Specialist in LLM Applications & Computer Vision
Instructor at O'Reilly Media
Technical Writer & Content Creator

Conversable Agents

- Conversable:
  - Entity with a specific role that can pass messages to send and receive information to and from other conversable agents,
  - e.g., to start or continue a conversation.
- Customizable:
  - Each agent can be configured based on application-specific needs to display complex behavior in multi-agent conversations, using a mix of basic back-end types.

Conversable Agents

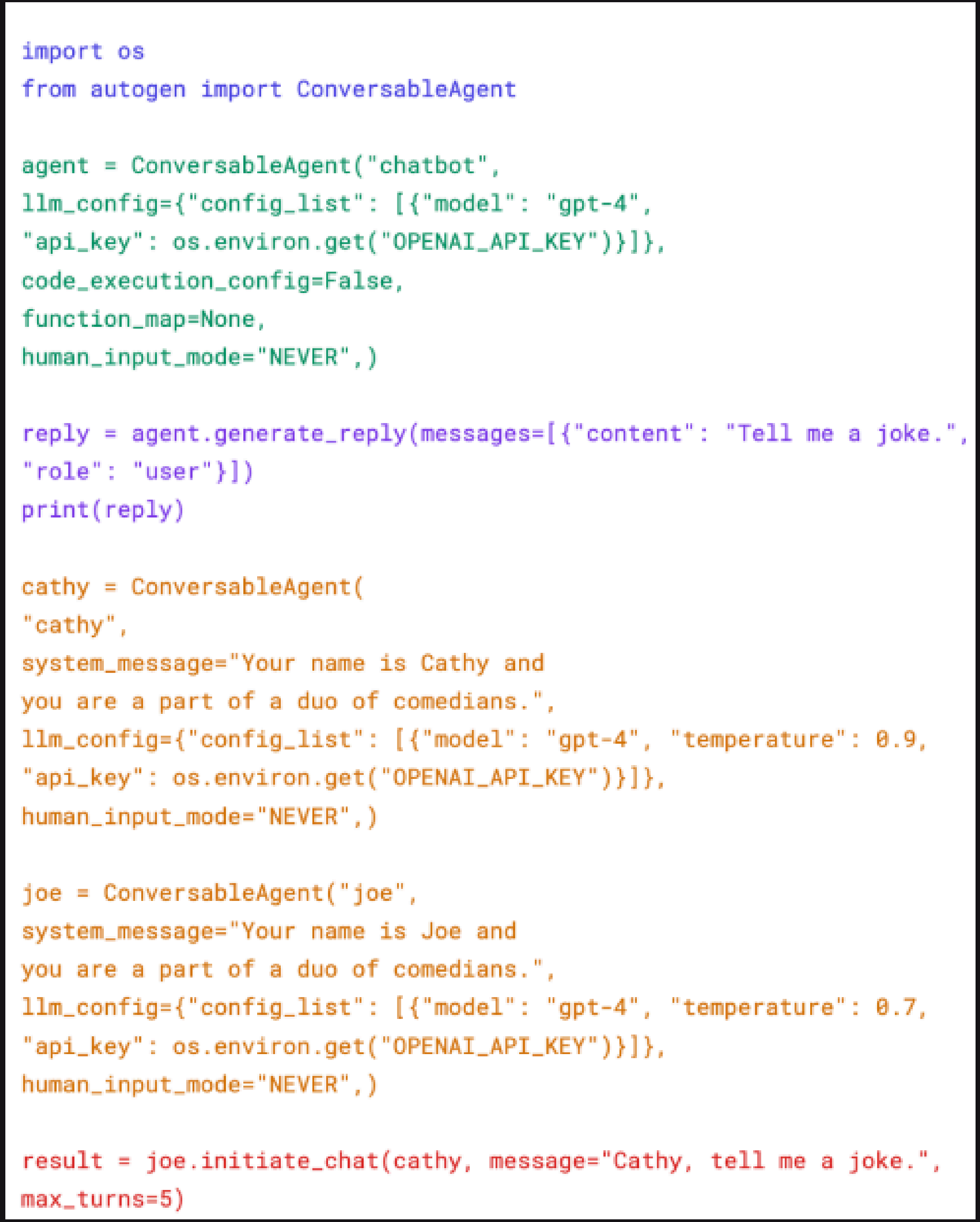
- Example



```
assistant = autogen.AssistantAgent(
            name="assistant",
            llm_config=llm_config,
            )
```

Conversable Agents

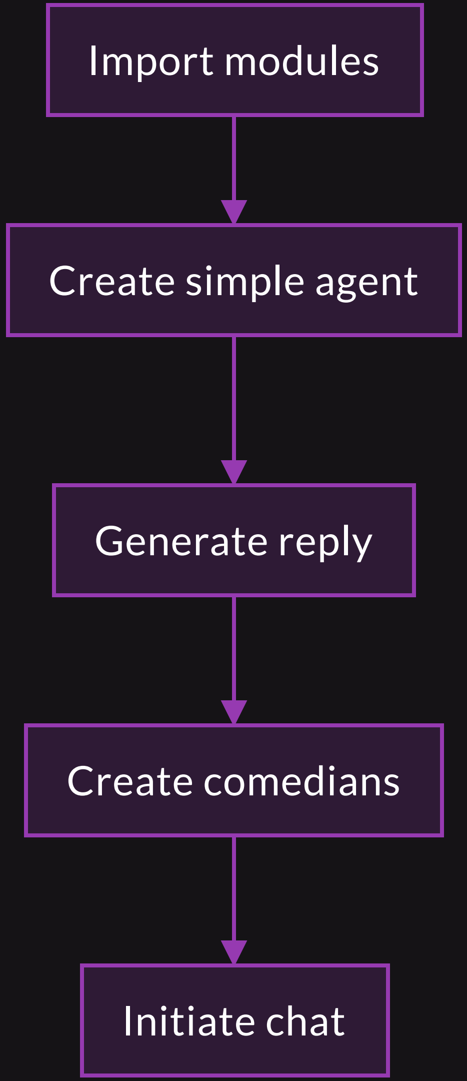
Code



```
import os
from autogen import ConversableAgent

agent = ConversableAgent("chatbot",
llm_config={"config_list": [{"model": "gpt-4",
"api_key": os.environ.get("OPENAI_API_KEY")}]},
code_execution_config=False,
function_map=None,
human_input_mode="NEVER",)

reply = agent.generate_reply(messages=[{"content": "Tell me a joke.",
"role": "user"}])
print(reply)

cathy = ConversableAgent(
"cathy",
system_message="Your name is Cathy and
you are a part of a duo of comedians.",
llm_config={"config_list": [{"model": "gpt-4", "temperature": 0.9,
"api_key": os.environ.get("OPENAI_API_KEY")}]},
human_input_mode="NEVER",)

joe = ConversableAgent("joe",
system_message="Your name is Joe and
you are a part of a duo of comedians.",
llm_config={"config_list": [{"model": "gpt-4", "temperature": 0.7,
"api_key": os.environ.get("OPENAI_API_KEY")}]},
human_input_mode="NEVER",)

result = joe.initiate_chat(cathy, message="Cathy, tell me a joke.",
max_turns=5)
```

Conversable Agents

Execution Flow



Import modules

Create simple agent

Generate reply

Create comedians

Initiate chat

Thank You