

```
import os
```

```
from swarms import Agent
```

```
from swarm_models import OpenAIChat
```

```
from swarms.structs.agents_available import showcase_available_agents
```

```
# Get the OpenAI API key from the environment variable
```

```
api_key = os.getenv("OPENAI_API_KEY")
```

```
# Create an instance of the OpenAIChat class
```

```
model = OpenAIChat(
```

```
    api_key=api_key, model_name="gpt-4o-mini", temperature=0.1
```

```
)
```

```
# Initialize the Claims Director agent
```

```
director_agent = Agent(
```

```
    agent_name="ClaimsDirector",
```

```
    agent_description="Oversees and coordinates the medical insurance claims processing  
workflow",
```

```
    system_prompt="""You are the Claims Director responsible for managing the medical insurance  
claims process.
```

```
    Assign and prioritize tasks between claims processors and auditors. Ensure claims are handled  
efficiently
```

```
    and accurately while maintaining compliance with insurance policies and regulations.""",
```

```
    llm=model,
```

```
max_loops=1,  
dashboard=False,  
streaming_on=True,  
verbose=True,  
stopping_token="<DONE>",  
state_save_file_type="json",  
saved_state_path="director_agent.json",  
)
```

# Initialize Claims Processor agent

```
processor_agent = Agent(  
    agent_name="ClaimsProcessor",  
    agent_description="Reviews and processes medical insurance claims, verifying coverage and  
eligibility",  
    system_prompt="""Review medical insurance claims for completeness and accuracy. Verify  
patient eligibility,  
coverage details, and process claims according to policy guidelines. Flag any claims requiring  
special review.""",  
    llm=model,  
    max_loops=1,  
    dashboard=False,  
    streaming_on=True,  
    verbose=True,  
    stopping_token="<DONE>",  
    state_save_file_type="json",  
    saved_state_path="processor_agent.json",
```

)

# Initialize Claims Auditor agent

auditor\_agent = Agent(

agent\_name="ClaimsAuditor",

agent\_description="Audits processed claims for accuracy and compliance with policies and regulations",

system\_prompt="""Audit processed insurance claims for accuracy and compliance. Review claim decisions,

identify potential fraud or errors, and ensure all processing follows established guidelines and regulations."""

llm=model,

max\_loops=1,

dashboard=False,

streaming\_on=True,

verbose=True,

stopping\_token="<DONE>",

state\_save\_file\_type="json",

saved\_state\_path="auditor\_agent.json",

)

# Create a list of agents

agents = [director\_agent, processor\_agent, auditor\_agent]

print(showcase\_available\_agents(agents=agents))