

```
from swarms import Agent
```

```
from swarm_models import OpenAIChat
```

```
def calculate_profit(revenue: float, expenses: float):
```

```
    """
```

```
    Calculates the profit by subtracting expenses from revenue.
```

```
    Args:
```

```
        revenue (float): The total revenue.
```

```
        expenses (float): The total expenses.
```

```
    Returns:
```

```
        float: The calculated profit.
```

```
    """
```

```
    return revenue - expenses
```

```
def generate_report(company_name: str, profit: float):
```

```
    """
```

```
    Generates a report for a company's profit.
```

```
    Args:
```

```
        company_name (str): The name of the company.
```

```
        profit (float): The calculated profit.
```

Returns:

str: The report for the company's profit.

"""

return f"The profit for {company_name} is \${profit}."

def account_agent(task: str = None):

"""

delegate a task to an agent!

Task: str (What task to give to an agent)

"""

agent = Agent(

agent_name="Finance Agent",

system_prompt="You're the Finance agent, your purpose is to generate a profit report for a company!",

agent_description="Generate a profit report for a company!",

llm=OpenAIChat(),

max_loops=1,

autosave=True,

dynamic_temperature_enabled=True,

dashboard=False,

verbose=True,

interactive=True, # Set to False to disable interactive mode

stopping_token="<DONE>",

```
# saved_state_path="accounting_agent.json",  
  
# tools=[calculate_profit, generate_report],  
  
# docs_folder="docs",  
  
# pdf_path="docs/accounting_agent.pdf",  
  
)
```

```
out = agent.run(task)
```

```
return out
```

```
# Initialize the agent
```

```
agent = Agent(  
    agent_name="Accounting Assistant",  
    system_prompt="You're the accounting agent, your purpose is to generate a profit report for a  
company!",  
    agent_description="Generate a profit report for a company!",  
    llm=OpenAIChat(),  
    max_loops=1,  
    autosave=True,  
    dynamic_temperature_enabled=True,  
    dashboard=False,  
    verbose=True,  
    # interactive=True, # Set to False to disable interactive mode  
    # stopping_token="<DONE>",  
    # saved_state_path="accounting_agent.json",  
    tools=[account_agent],
```

```
# docs_folder="docs",  
  
# pdf_path="docs/accounting_agent.pdf",  
  
)  
  
agent.run(  
    "Delegate a task to the accounting agent: what are the best ways to read cashflow statements"  
)
```