

```
!pip install langgraph langsmith langchain_groq langchain_community

/Local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (1.6.0)
sr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (6.4.4)
local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (0.3.1)
/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain_community) (1.20.0)
json<0.7,>=0.5.7->langchain_community)
ta (7.3 kB)
on<0.7,>=0.5.7->langchain_community)
data (1.5 kB)
ocal/lib/python3.11/dist-packages (from groq<1,>=0.4.1->langchain_groq) (4.9.0)
local/lib/python3.11/dist-packages (from groq<1,>=0.4.1->langchain_groq) (1.9.0)
/python3.11/dist-packages (from groq<1,>=0.4.1->langchain_groq) (1.3.1)
0 in /usr/local/lib/python3.11/dist-packages (from groq<1,>=0.4.1->langchain_groq) (4.13.2)
/python3.11/dist-packages (from httpx<1,>=0.23.0->langsmith) (2025.4.26)
al/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->langsmith) (1.0.9)
thon3.11/dist-packages (from httpx<1,>=0.23.0->langsmith) (3.10)
ib/python3.11/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->langsmith) (0.16.0)
1.0.0,>=0.3.8 in /usr/local/lib/python3.11/dist-packages (from langchain<1.0.0,>=0.3.25->langchain_community) (0.3.8)
usr/local/lib/python3.11/dist-packages (from langchain-core>=0.1->langgraph) (1.33)
point>=2.0.26->langgraph)
7_x86_64.manylinux2014_x86_64.whl.metadata (43 kB)
kB 3.0 MB/s eta 0:00:00
/usr/local/lib/python3.11/dist-packages (from pydantic>=2.7.4->langgraph) (0.7.0)
/usr/local/lib/python3.11/dist-packages (from pydantic>=2.7.4->langgraph) (2.33.2)
in /usr/local/lib/python3.11/dist-packages (from pydantic>=2.7.4->langgraph) (0.4.1)
<3.0.0,>=2.4.0->langchain_community)
ata (24 kB)
in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2->langsmith) (3.4.2)
r/local/lib/python3.11/dist-packages (from requests<3,>=2->langsmith) (2.4.0)
/lib/python3.11/dist-packages (from SQLAlchemy<3,>=1.4->langchain_community) (3.2.2)
local/lib/python3.11/dist-packages (from jsonpatch<2.0,>=1.33->langchain-core>=0.1->langgraph) (3.0.0)
,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langchain_community)
adata (1.1 kB)

kB 12.2 MB/s eta 0:00:00
)
(2.5 MB)
2.9 MB/s eta 0:00:00
kB)

kB 13.4 MB/s eta 0:00:00

(44 kB)
3.8 MB/s eta 0:00:00
3 kB)
)
4.3 MB/s eta 0:00:00
kB)
3.6 MB/s eta 0:00:00

4.5 MB/s eta 0:00:00
x86_64.manylinux2014_x86_64.whl (216 kB)
kB 15.8 MB/s eta 0:00:00

B)
kB)
mypy-extensions, marshmallow, httpx-sse, typing-inspect, pydantic-settings, langgraph-sdk, groq, dataclasses-json, langgraph-checkpoint,
0 httpx-sse-0.4.0 langchain_community-0.3.24 langchain_groq-0.3.2 langgraph-0.4.7 langgraph-checkpoint-2.0.26 langgraph-prebuilt-0.2.2
```

```
from langchain_groq import ChatGroq
from typing import Annotated
from typing_extensions import TypedDict
from langgraph.graph import StateGraph, START, END
from langgraph.graph.message import add_messages
from IPython.display import Image, display
```

```
groq_api_key = "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"
```

```
llm = ChatGroq(groq_api_key=groq_api_key, model="Gemma2-9b-it")
```

```
class State(TypedDict):
    messages: Annotated[list, add_messages]
    sentiment: str
```

```
def preprocess(state: State) -> State:
    cleaned = state['messages'][-1].content.strip()
    state['messages'][-1].content = cleaned
```

```

state["messages"] = messages
return state

def analyze_sentiment(state: State) -> State:
    msg = state["messages"][-1].content
    state["sentiment"] = 'positive' if 'good' in msg else 'neutral'
    return state

def chatbot(state: State) -> State:
    return {'messages': llm.invoke(state['messages'])}

def logger(state: State) -> State:
    print(f'LOG: {state["messages"][-1].content}, Sentiment: {state.get("sentiment")}')
    return state

```

```

builder = StateGraph(State)
builder.add_node('preprocess', preprocess)
builder.add_node('analyze_sentiment', analyze_sentiment)
builder.add_node('chatbot', chatbot)
builder.add_node('logger', logger)

↳ <langgraph.graph.state.StateGraph at 0x79e86c8b79d0>

```

```

builder.add_edge(START, 'preprocess')
builder.add_edge('preprocess', 'analyze_sentiment')
builder.add_edge('analyze_sentiment', 'chatbot')
builder.add_edge('chatbot', 'logger')
builder.add_edge('logger', END)

↳ <langgraph.graph.state.StateGraph at 0x79e86c8b79d0>

```

```
builder.compile()
```



Start coding or [generate](#) with AI.