

[Tutorials](#)[Build an Agent](#)

Build an Agent

[Open in Colab](#)[Open on GitHub](#)

By themselves, language models can't take actions - they just output text. A big use case for LangChain is creating **agents**. Agents are systems that use LLMs as reasoning engines to determine which actions to take and the inputs necessary to perform the action. After executing actions, the results can be fed back into the LLM to determine whether more actions are needed, or whether it is okay to finish. This is often achieved via tool-calling.

In this tutorial we will build an agent that can interact with a search engine. You will be able to ask this agent questions, watch it call the search tool, and have conversations with it.

End-to-end agent

The code snippet below represents a fully functional agent that uses an LLM to decide which tools to use. It is equipped with a generic

search tool. It has conversational memory - meaning that it can be used as a multi-turn chatbot.

In the rest of the guide, we will walk through the individual components and what each part does - but if you want to just grab some code and get started, feel free to use this!

```
# Import relevant functionality
from langchain_anthropic import ChatAnthropic
from langchain_community.tools.tavily_search import
TavilySearchResults
from langchain_core.messages import HumanMessage
from langgraph.checkpoint.memory import MemorySaver
from langgraph.prebuilt import create_react_agent

# Create the agent
memory = MemorySaver()
model = ChatAnthropic(model_name="claude-3-sonnet-
20240229")
search = TavilySearchResults(max_results=2)
tools = [search]
agent_executor = create_react_agent(model, tools,
checkpointer=memory)
```

API Reference: [ChatAnthropic](#) | [TavilySearchResults](#) |

[HumanMessage](#) | [MemorySaver](#) | [create_react_agent](#)

```
# Use the agent
config = {"configurable": {"thread_id": "abc123"}}
for step in agent_executor.stream(
    {"messages": [HumanMessage(content="hi im bob!
and i live in sf")]},
    config,
    stream_mode="values",
):
    step["messages"][-1].pretty_print()
```

```
===== [1m Human Message
[0m=====
```

```
hi im bob! and i live in sf
```

```
===== [1m Ai Message
[0m=====
```

Hello Bob! Since you didn't ask a specific question, I don't need to use any tools right now. I'm an AI assistant created by Anthropic to be helpful, honest, and harmless. Feel free to ask me anything and I'll do my best to provide a useful response or look up information using my capabilities.

```
for step in agent_executor.stream(
    {"messages": [HumanMessage(content="whats the
```

```

weather where I live?"]]},
    config,
    stream_mode="values",
):
    step["messages"][-1].pretty_print()

```

```

===== [1m Human Message
 [0m=====

```

whats the weather where I live?

```

===== [1m Ai Message
 [0m=====

```

```

[{'text': 'To get the current weather for your
location in San Francisco, I can use the
tavily_search_results_json tool:', 'type': 'text'},
{'id': 'toolu_01AKa2MErG1CU3zRiGsvpBud', 'input':
{'query': 'san francisco weather'}, 'name':
'tavily_search_results_json', 'type': 'tool_use'}]

```

Tool Calls:

```

    tavily_search_results_json
(toolu_01AKa2MErG1CU3zRiGsvpBud)
    Call ID: toolu_01AKa2MErG1CU3zRiGsvpBud
    Args:

```

```

        query: san francisco weather

```

```

===== [1m Tool Message
 [0m=====

```

```

Name: tavily_search_results_json

```

```
[{"url": "https://www.weatherapi.com/", "content": "  
{  
'location': {  
'name': 'San Francisco', 'region':  
'California', 'country': 'United States of America',  
'lat': 37.775, 'lon': -122.4183, 'tz_id':  
'America/Los_Angeles', 'localtime_epoch': 1739994486,  
'localtime': '2025-02-19 11:48'}, 'current':  
{  
'last_updated_epoch': 1739994300, 'last_updated':  
'2025-02-19 11:45', 'temp_c': 13.3, 'temp_f': 55.9,  
'is_day': 1, 'condition': {'text': 'Light rain',  
'icon':  
'//cdn.weatherapi.com/weather/64x64/day/296.png',  
'code': 1183}, 'wind_mph': 5.8, 'wind_kph': 9.4,  
'wind_degree': 195, 'wind_dir': 'SSW', 'pressure_mb':  
1023.0, 'pressure_in': 30.2, 'precip_mm': 0.0,  
'precip_in': 0.0, 'humidity': 87, 'cloud': 100,  
'feelslike_c': 12.7, 'feelslike_f': 54.8,  
'windchill_c': 9.1, 'windchill_f': 48.4,  
'heatindex_c': 10.2, 'heatindex_f': 50.3,  
'dewpoint_c': 9.8, 'dewpoint_f': 49.7, 'vis_km': 4.0,  
'vis_miles': 2.0, 'uv': 1.4, 'gust_mph': 8.9,  
'gust_kph': 14.4}}"}, {"url": "https://world-  
weather.info/forecast/usa/san_francisco/february-  
2025/", "content": "Weather in San Francisco in  
February 2025 (California) – Detailed Weather  
Forecast for a Month Weather World Weather in San  
Francisco Weather in San Francisco in February 2025  
San Francisco Weather Forecast for February 2025, is  
based on previous years' statistical data. +59°+50°  
+59°+52° +59°+50° +61°+52° +59°+50° +61°+50° +61°+52°  
+63°+52° +61°+52° +61°+50° +61°+50° +61°+50° +59°+50°
```

```
+59°+50° +61°+50° +61°+52° +59°+50° +59°+48° +57°+48°
+59°+50° +59°+48° +59°+50° +57°+46° +61°+50° +61°+50°
+59°+50° +59°+48° +59°+50° Extended weather forecast
in San Francisco HourlyWeek10-Day14-Day30-DayYear
Weather in large and nearby cities Weather in
Washington, D.C.+41° Sacramento+55° Pleasanton+55°
Redwood City+55° San Leandro+55° San Mateo+54° San
Rafael+52° San Ramon+52° South San Francisco+54°
Vallejo+50° Palo Alto+55° Pacifica+55° Berkeley+54°
Castro Valley+55° Concord+52° Daly City+54°
Noverd+52° Sign Hill+54° world's temperature today
day day Temperature units"}]
===== [1m Ai Message
[0m=====
```

The search results provide the current weather conditions and forecast for San Francisco. According to the data from WeatherAPI, the current temperature in San Francisco is around 55°F (13°C) with light rain and winds around 6 mph. The extended forecast shows temperatures ranging from the upper 40s to low 60s Fahrenheit over the next few weeks.

So in summary, it's a cool, rainy day currently in San Francisco where you live, Bob. Let me know if you need any other details about the weather there!

Setup

Jupyter Notebook

This guide (and most of the other guides in the documentation) uses [Jupyter notebooks](#) and assumes the reader is as well. Jupyter notebooks are perfect interactive environments for learning how to work with LLM systems because oftentimes things can go wrong (unexpected output, API down, etc), and observing these cases is a great way to better understand building with LLMs.

This and other tutorials are perhaps most conveniently run in a Jupyter notebook. See [here](#) for instructions on how to install.

Installation

To install LangChain run:

```
%pip install -U langchain-community langgraph  
langchain-anthropic tavily-python langgraph-  
checkpoint-sqlite
```

For more details, see our [Installation guide](#).

LangSmith

Many of the applications you build with LangChain will contain multiple steps with multiple invocations of LLM calls. As these applications get more and more complex, it becomes crucial to be able to inspect what exactly is going on inside your chain or agent. The best way to do this is with [LangSmith](#).

After you sign up at the link above, make sure to set your environment variables to start logging traces:

```
export LANGSMITH_TRACING="true"
export LANGSMITH_API_KEY="..."
```

Or, if in a notebook, you can set them with:

```
import getpass
import os

os.environ["LANGSMITH_TRACING"] = "true"
os.environ["LANGSMITH_API_KEY"] = getpass.getpass()
```

Tavily

We will be using **Tavily** (a search engine) as a tool. In order to use it, you will need to get and set an API key:

```
export TAVILY_API_KEY="..."
```

Or, if in a notebook, you can set it with:

```
import getpass
import os

os.environ["TAVILY_API_KEY"] = getpass.getpass()
```

Define tools

We first need to create the tools we want to use. Our main tool of choice will be **Tavily** - a search engine. We have a built-in tool in LangChain to easily use Tavily search engine as tool.

```
from langchain_community.tools.tavily_search import
TavilySearchResults

search = TavilySearchResults(max_results=2)
```

```
search_results = search.invoke("what is the weather  
in SF")  
print(search_results)  
# If we want, we can create other tools.  
# Once we have all the tools we want, we can put  
them in a list that we will reference later.  
tools = [search]
```

API Reference: TavilySearchResults

```
[{'url': 'https://www.weatherapi.com/', 'content': "  
{  
  'location': {  
    'name': 'San Francisco',  
    'region': 'California',  
    'country': 'United States of America',  
    'lat': 37.775,  
    'lon': -122.4183,  
    'tz_id': 'America/Los_Angeles',  
    'localtime_epoch': 1739993250,  
    'localtime': '2025-02-19 11:27'},  
    'current': {  
      'last_updated_epoch': 1739992500,  
      'last_updated': '2025-02-19 11:15',  
      'temp_c': 13.3,  
      'temp_f': 55.9,  
      'is_day': 1,  
      'condition': {  
        'text': 'Light rain',  
        'icon':  
        'https://cdn.weatherapi.com/weather/64x64/day/296.png',  
        'code': 1183},  
      'wind_mph': 5.8,  
      'wind_kph': 9.4,  
      'wind_degree': 195,  
      'wind_dir': 'SSW',  
      'pressure_mb': 1023.0,  
      'pressure_in': 30.2,  
      'precip_mm': 0.0,  
      'precip_in': 0.0,  
      'humidity': 87,  
      'cloud': 100,  
      'feelslike_c': 12.7,  
      'feelslike_f': 54.8,  
      'windchill_c': 9.1,  
      'windchill_f': 48.4,  
      'heatindex_c': 10.2,  
      'heatindex_f': 50.3,
```

```
'dewpoint_c': 9.8, 'dewpoint_f': 49.7, 'vis_km': 4.0,  
'vis_miles': 2.0, 'uv': 1.4, 'gust_mph': 8.9,  
'gust_kph': 14.4}}"}}, {'url':  
'https://weathershgun.com/weather/usa/ca/san-  
francisco/480/february/2025-02-19', 'content': 'San  
Francisco, California Weather: Wednesday, February  
19, 2025. Cloudy weather, overcast skies with clouds.  
Day 61°. Night 43°.'}]
```

Using Language Models

Next, let's learn how to use a language model to call tools. LangChain supports many different language models that you can use interchangeably - select the one you want to use below!

Select **chat model**:

OpenAI ▾

```
pip install -qU "langchain[openai]"
```

```
import getpass  
import os  
  
if not os.environ.get("OPENAI_API_KEY"):
```

```
os.environ["OPENAI_API_KEY"] =  
getpass.getpass("Enter API key for OpenAI: ")  
  
from langchain.chat_models import init_chat_model  
  
model = init_chat_model("gpt-4",  
model_provider="openai")
```

You can call the language model by passing in a list of messages. By default, the response is a `content` string.

```
from langchain_core.messages import HumanMessage  
  
response =  
model.invoke([HumanMessage(content="hi!")])  
response.content
```

API Reference: [HumanMessage](#)

```
'Hi there!'
```

We can now see what it is like to enable this model to do tool calling. In order to enable that we use `.bind_tools` to give the language model knowledge of these tools

```
model_with_tools = model.bind_tools(tools)
```

We can now call the model. Let's first call it with a normal message, and see how it responds. We can look at both the `content` field as well as the `tool_calls` field.

```
response =  
model_with_tools.invoke([HumanMessage(content="Hi!")])  
  
print(f"ContentString: {response.content}")  
print(f"ToolCalls: {response.tool_calls}")
```

```
ContentString: Hello!  
ToolCalls: []
```

Now, let's try calling it with some input that would expect a tool to be called.

```
response =  
model_with_tools.invoke([HumanMessage(content="What's  
the weather in SF?")])
```

```
print(f"ContentString: {response.content}")  
print(f"ToolCalls: {response.tool_calls}")
```

```
ContentString:  
ToolCalls: [{'name': 'tavily_search_results_json',  
  'args': {'query': 'weather san francisco'}, 'id':  
  'toolu_01VTP7DUvSfgtYxsq9x4EwMp'}]
```

We can see that there's now no text content, but there is a tool call! It wants us to call the Tavily Search tool.

This isn't calling that tool yet -it's just telling us to. In order to actually call it, we'll want to create our agent.

Create the agent

Now that we have defined the tools and the LLM, we can create the agent. We will be using [LangGraph](#) to construct the agent. Currently, we are using a high level interface to construct the agent, but the nice thing about LangGraph is that this high-level interface is backed by a low-level, highly controllable API in case you want to modify the

agent logic.

Now, we can initialize the agent with the LLM and the tools.

Note that we are passing in the `model`, not `model_with_tools`. That is because `create_react_agent` will call `.bind_tools` for us under the hood.

```
from langgraph.prebuilt import create_react_agent  
  
agent_executor = create_react_agent(model, tools)
```

API Reference: [create_react_agent](#)

Run the agent

We can now run the agent with a few queries! Note that for now, these are all **stateless** queries (it won't remember previous interactions). Note that the agent will return the **final** state at the end of the interaction (which includes any inputs, we will see later on how to get only the outputs).

First up, let's see how it responds when there's no need to call a tool:

```
response = agent_executor.invoke({"messages":  
[HumanMessage(content="hi!")])  
  
response["messages"]
```

```
[HumanMessage(content='hi!', id='a820fcc5-9b87-457a-  
9af0-f21768143ee3'),  
  AIMessage(content='Hello!', response_metadata={'id':  
'msg_01VbC493X1VEDyusgttiEr1z', 'model': 'claude-3-  
sonnet-20240229', 'stop_reason': 'end_turn',  
'stop_sequence': None, 'usage': {'input_tokens': 264,  
'output_tokens': 5}}, id='run-0e0ddae8-a85b-4bd6-  
947c-c36c857a4698-0', usage_metadata={'input_tokens':  
264, 'output_tokens': 5, 'total_tokens': 269})]
```

In order to see exactly what is happening under the hood (and to make sure it's not calling a tool) we can take a look at the [LangSmith trace](#)

Let's now try it out on an example where it should be invoking the tool

```
response = agent_executor.invoke(  
    [{"messages":  
        [HumanMessage(content="What's the weather like in San Francisco?")]  
    }]
```



```
        {"messages": [HumanMessage(content="whats the  
weather in sf?")]}  
    )  
    response["messages"]
```

```
[HumanMessage(content='whats the weather in sf?',  
id='1d6c96bb-4ddb-415c-a579-a07d5264de0d'),  
  AIMessage(content=[{'id':  
    'toolu_01Y5EK4bw2LqsQXeaUv8iueF', 'input': {'query':  
    'weather in san francisco'}, 'name':  
    'tavily_search_results_json', 'type': 'tool_use'}],  
    response_metadata={'id':  
    'msg_0132wQUcEduJ8UKVVVqwJzM4', 'model': 'claude-3-  
sonnet-20240229', 'stop_reason': 'tool_use',  
    'stop_sequence': None, 'usage': {'input_tokens': 269,  
    'output_tokens': 61}}, id='run-26d5e5e8-d4fd-46d2-  
a197-87b95b10e823-0', tool_calls=[{'name':  
    'tavily_search_results_json', 'args': {'query':  
    'weather in san francisco'}, 'id':  
    'toolu_01Y5EK4bw2LqsQXeaUv8iueF'}], usage_metadata=  
    {'input_tokens': 269, 'output_tokens': 61,  
    'total_tokens': 330}),  
  ToolMessage(content='[{"url":  
    "https://www.weatherapi.com/", "content": "  
    {\n      \'location\': {\n        \'name\': \'San Francisco\',  
        \'region\': \'California\',  
        \'country\': \'United States of America\',  
        \'lat\': 37.78,  
        \'lon\': -122.42,  
        \'tz_id\': \'America/Los_Angeles\',  
        \'localtime_epoch\': 1717238703,  
        \'localtime\':
```

```
\ '2024-06-01 3:45\ '}, \ 'current\ ':  
{\ 'last_updated_epoch\ ': 1717237800,  
\ 'last_updated\ ': \ '2024-06-01 03:30\ ', \ 'temp_c\ ':  
12.0, \ 'temp_f\ ': 53.6, \ 'is_day\ ': 0, \ 'condition\ ':  
{\ 'text\ ': \ 'Mist\ ', \ 'icon\ ':  
\ '//cdn.weatherapi.com/weather/64x64/night/143.png\ ',  
\ 'code\ ': 1030}, \ 'wind_mph\ ': 5.6, \ 'wind_kph\ ':  
9.0, \ 'wind_degree\ ': 310, \ 'wind_dir\ ': \ 'NW\ ',  
\ 'pressure_mb\ ': 1013.0, \ 'pressure_in\ ': 29.92,  
\ 'precip_mm\ ': 0.0, \ 'precip_in\ ': 0.0, \ 'humidity\ ':  
88, \ 'cloud\ ': 100, \ 'feelslike_c\ ': 10.5,  
\ 'feelslike_f\ ': 50.8, \ 'windchill_c\ ': 9.3,  
\ 'windchill_f\ ': 48.7, \ 'heatindex_c\ ': 11.1,  
\ 'heatindex_f\ ': 51.9, \ 'dewpoint_c\ ': 8.8,  
\ 'dewpoint_f\ ': 47.8, \ 'vis_km\ ': 6.4, \ 'vis_miles\ ':  
3.0, \ 'uv\ ': 1.0, \ 'gust_mph\ ': 12.5, \ 'gust_kph\ ':  
20.1}}"}}, {"url":  
"https://www.timeanddate.com/weather/usa/san-  
francisco/hourly", "content": "Sun & Moon. Weather  
Today Weather Hourly 14 Day Forecast Yesterday/Past  
Weather Climate (Averages) Currently: 59 \u00b0F.  
Passing clouds. (Weather station: San Francisco  
International Airport, USA). See more current  
weather."}}]', name='tavily_search_results_json',  
id='37aa1fd9-b232-4a02-bd22-bc5b9b44a22c',  
tool_call_id='toolu_01Y5EK4bw2LqsQXeaUv8iueF'),  
  AIMessage(content='Based on the search results, here  
is a summary of the current weather in San  
Francisco:\n\nThe weather in San Francisco is  
currently misty with a temperature of around 53°F
```

```
(12°C). There is complete cloud cover and moderate winds from the northwest around 5–9 mph (9–14 km/h). Humidity is high at 88%. Visibility is around 3 miles (6.4 km). \n\nThe results provide an hourly forecast as well as current conditions from a couple different weather sources. Let me know if you need any additional details about the San Francisco weather!', response_metadata={'id': 'msg_01BRX9mrT19nBDdHYtR7wJ92', 'model': 'claude-3-sonnet-20240229', 'stop_reason': 'end_turn', 'stop_sequence': None, 'usage': {'input_tokens': 920, 'output_tokens': 132}}, id='run-d0325583-3ddc-4432-b2b2-d023eb97660f-0', usage_metadata={'input_tokens': 920, 'output_tokens': 132, 'total_tokens': 1052}}])
```

We can check out the [LangSmith trace](#) to make sure it's calling the search tool effectively.

Streaming Messages

We've seen how the agent can be called with `.invoke` to get a final response. If the agent executes multiple steps, this may take a while. To show intermediate progress, we can stream back messages as they occur.

```

for step in agent_executor.stream(
    {"messages": [HumanMessage(content="whats the
weather in sf?")]}],
    stream_mode="values",
):
    step["messages"][-1].pretty_print()

```

```

===== [1m Human Message
[0m=====

```

whats the weather in sf?

```

===== [1m Ai Message
[0m=====

```

```

[{'text': 'Okay, let me look up the current weather
for San Francisco using a search engine:', 'type':
'text'}, {'id': 'toolu_01H1brh5EZpZqtqHBxkosPtN',
'input': {'query': 'san francisco weather'}, 'name':
'tavily_search_results_json', 'type': 'tool_use'}]

```

Tool Calls:

```

    tavily_search_results_json
(toolu_01H1brh5EZpZqtqHBxkosPtN)
Call ID: toolu_01H1brh5EZpZqtqHBxkosPtN
Args:
    query: san francisco weather

```

```

===== [1m Tool Message
[0m=====

```

Name: tavily_search_results_json

```
[{"url": "https://www.weatherapi.com/", "content": "  
{ 'location': { 'name': 'San Francisco', 'region':  
'California', 'country': 'United States of America',  
'lat': 37.775, 'lon': -122.4183, 'tz_id':  
'America/Los_Angeles', 'localtime_epoch': 1739994486,  
'localtime': '2025-02-19 11:48'}, 'current':  
{ 'last_updated_epoch': 1739994300, 'last_updated':  
'2025-02-19 11:45', 'temp_c': 13.3, 'temp_f': 55.9,  
'is_day': 1, 'condition': { 'text': 'Light rain',  
'icon':  
'//cdn.weatherapi.com/weather/64x64/day/296.png',  
'code': 1183}, 'wind_mph': 5.8, 'wind_kph': 9.4,  
'wind_degree': 195, 'wind_dir': 'SSW', 'pressure_mb':  
1023.0, 'pressure_in': 30.2, 'precip_mm': 0.0,  
'precip_in': 0.0, 'humidity': 87, 'cloud': 100,  
'feelslike_c': 12.7, 'feelslike_f': 54.8,  
'windchill_c': 9.1, 'windchill_f': 48.4,  
'heatindex_c': 10.2, 'heatindex_f': 50.3,  
'dewpoint_c': 9.8, 'dewpoint_f': 49.7, 'vis_km': 4.0,  
'vis_miles': 2.0, 'uv': 1.4, 'gust_mph': 8.9,  
'gust_kph': 14.4}}"}, {"url": "https://world-  
weather.info/forecast/usa/san_francisco/february-  
2025/", "content": "Weather in San Francisco in  
February 2025 (California) – Detailed Weather  
Forecast for a Month Weather World Weather in San  
Francisco Weather in San Francisco in February 2025  
San Francisco Weather Forecast for February 2025, is  
based on previous years' statistical data. +59°+50°  
+59°+52° +59°+50° +61°+52° +59°+50° +61°+50° +61°+52°
```

+63°+52° +61°+52° +61°+50° +61°+50° +61°+50° +59°+50°
+59°+50° +61°+50° +61°+52° +59°+50° +59°+48° +57°+48°
+59°+50° +59°+48° +59°+50° +57°+46° +61°+50° +61°+50°
+59°+50° +59°+48° +59°+50° Extended weather forecast
in San Francisco HourlyWeek10–Day14–Day30–DayYear
Weather in large and nearby cities Weather in
Washington, D.C.+41° Sacramento+55° Pleasanton+55°
Redwood City+55° San Leandro+55° San Mateo+54° San
Rafael+52° San Ramon+52° South San Francisco+54°
Vallejo+50° Palo Alto+55° Pacifica+55° Berkeley+54°
Castro Valley+55° Concord+52° Daly City+54°
Noverd+52° Sign Hill+54° world's temperature today
day day Temperature units"}]
===== [1m Ai Message
[0m=====

The search results provide details on the current weather conditions and forecast for San Francisco. Some key details:

- It is lightly raining in San Francisco right now, with a temperature around 55°F/13°C.
- The forecast for the rest of February 2025 shows daytime highs mostly in the upper 50s to low 60s F, with night lows in the upper 40s to low 50s F.
- Typical weather includes some rain, clouds, cool temperatures and breezy conditions.

So in summary, as is common for San Francisco in late winter, it is currently cool with light rain showers,

and similar mild, unsettled weather is expected over the next couple weeks. Layers and a light jacket would be advisable for being outdoors. Let me know if you need any other details!

Streaming tokens

In addition to streaming back messages, it is also useful to stream back tokens. We can do this by specifying

```
stream_mode="messages".
```

::: note

Below we use `message.text()`, which requires `langchain-core>=0.3.37`.

:::

```
for step, metadata in agent_executor.stream(
    {"messages": [HumanMessage(content="whats the
weather in sf?")]},
    stream_mode="messages",
):
    if metadata["langgraph_node"] == "agent" and
```

```
(text := step.text()):  
    print(text, end="|")
```

Based on the weather search results, here are the key details about the weather in San Francisco:

- The current temperature in San Francisco is around 55-56°F (13°C). Light rain is occurring with 100% cloud cover.

- Winds are around 5-9 mph from the south-southwest.

- The forecast for the rest of February 2025 shows daytime highs mostly in the upper 50s to low 60s°F, with overnight lows in the upper 40s to low 50s°F.

- Overall, typical cool and showery late winter weather is expected in San Francisco for the remainder of February, with a mix of rain and dry periods. Temperatures will be seasonable for this time of year.

So in summary, San Francisco is experiencing light rain and cool temperatures currently, but the late winter forecast shows typical mild and


```
show|ery conditions| pers|isting through the en|d of  
the| month.| Let| me know if you| need any other|  
details about| the weather in the| city!|
```

Adding in memory

As mentioned earlier, this agent is stateless. This means it does not remember previous interactions. To give it memory we need to pass in a checkpointer. When passing in a checkpointer, we also have to pass in a `thread_id` when invoking the agent (so it knows which thread/conversation to resume from).

```
from langgraph.checkpoint.memory import MemorySaver  
  
memory = MemorySaver()
```

API Reference: [MemorySaver](#)

```
agent_executor = create_react_agent(model, tools,  
    checkpointer=memory)  
  
config = {"configurable": {"thread_id": "abc123"}}
```

```

for chunk in agent_executor.stream(
    {"messages": [HumanMessage(content="hi im
bob!")]}], config
):
    print(chunk)
    print("----")

```

```

{'agent': {'messages': [AIMessage(content="Hello Bob!
It's nice to meet you again.", response_metadata=
{'id': 'msg_013C1z2ZySagEFwmU1EsysR2', 'model':
'claude-3-sonnet-20240229', 'stop_reason':
'end_turn', 'stop_sequence': None, 'usage':
{'input_tokens': 1162, 'output_tokens': 14}}],
id='run-f878acfd-d195-44e8-9166-e2796317e3f8-0',
usage_metadata={'input_tokens': 1162,
'output_tokens': 14, 'total_tokens': 1176})]}}
----

```

```

for chunk in agent_executor.stream(
    {"messages": [HumanMessage(content="whats my
name?")]}], config
):
    print(chunk)
    print("----")

```

```

{'agent': {'messages': [AIMessage(content="You

```

```
{ 'agent': { 'messages': [AIMessage(content= 'you
mentioned your name is Bob when you introduced
yourself earlier. So your name is Bob.',
response_metadata={'id':
'msg_01WNwnRNGwGDRw6vRdivt6i1', 'model': 'claude-3-
sonnet-20240229', 'stop_reason': 'end_turn',
'stop_sequence': None, 'usage': {'input_tokens':
1184, 'output_tokens': 21}}, id='run-f5c0b957-8878-
405a-9d4b-a7cd38efe81f-0', usage_metadata=
{'input_tokens': 1184, 'output_tokens': 21,
'total_tokens': 1205})]}}
```

Example [LangSmith trace](#)

If you want to start a new conversation, all you have to do is change the `thread_id` used

```
config = {"configurable": {"thread_id": "xyz123"}}
for chunk in agent_executor.stream(
    {"messages": [HumanMessage(content="whats my
name?")]}], config
):
    print(chunk)
    print("----")
```

```
{'agent': {'messages': [AIMessage(content="I'm afraid
```

```
I don't actually know your name. As an AI assistant
without personal information about you, I don't have
a specific name associated with our conversation.",
response_metadata={'id':
'msg_01NoaXNNYZKSoBncPcLkdcbo', 'model': 'claude-3-
sonnet-20240229', 'stop_reason': 'end_turn',
'stop_sequence': None, 'usage': {'input_tokens': 267,
'output_tokens': 36}}, id='run-c9f7df3d-525a-4d8f-
bbcf-a5b4a5d2e4b0-0', usage_metadata={'input_tokens':
267, 'output_tokens': 36, 'total_tokens': 303}})]}}
----
```

Conclusion

That's a wrap! In this quick start we covered how to create a simple agent. We've then shown how to stream back a response - not only with the intermediate steps, but also tokens! We've also added in memory so you can have a conversation with them. Agents are a complex topic with lots to learn!

For more information on Agents, please check out the [LangGraph](#) documentation. This has it's own set of concepts, tutorials, and how-to guides.

Was this page helpful?



18 comments · 19 replies – powered by [giscus](#)

Oldest

Newest



ganeshghag [Nov 24, 2024](#)

assuming there are multiple agents tools being bound into the model, how does "a qualitative input NLP question" trigger appropriate tool? there could be multiple tools bound which are capable of providing the answer?

↑ 1



2 replies



ZcyhFromC [Nov 28, 2024](#)

Hi gang. With multi tools available, agents answer the question in three steps:

1. Let llm leveraging the **metadata of tools** to reason out which tool to use. Some description will work.
 2. Let python code invoke the tool and feed the response back to llm
 3. llm final output
- The key is that llm will first act as a router.





ganeshghag [Dec 9, 2024](#)

Thanks I read in detail about "tool calling" and now realize your reply is significant



ganeshghag [Nov 26, 2024](#)

Also, how can I rely on the third-party tool implementations, to give my users proper responses.

I cannot be testing for all possible inputs to these tools.

With above ambiguity, I am relegated to writing agents and my own custom tools

Tools appear to be wrappers over remote REST API, with a technical interface like runnable so that the agent can auto-invoke these tools

its a meek attempt to standardize AI agents with public REST APIs

↑ 1



0 replies



shan-max [Nov 30, 2024](#)

yes i am there , and my name shanawaz.

↑ 1



0 replies



shaojun [Dec 10, 2024](#)

can you show how to use `trim_messages` to limit the messages in `Saver` in this agent scenario?

↑ 1 

1 reply



shan-max [Mar 4](#)

assuming there are multiple agents tools being bound into the model, how does "a qualitative input NLP question" trigger appropriate tool? there could be multiple tools bound which are capable of providing the answer?



Bhargav2525 [Dec 13, 2024](#)

I am not getting response. It seems llm is not using that tool.

```
response = agent_executor.invoke(  
{"messages": [HumanMessage(content="whats the weather in sf?")]}  
)  
response["messages"]  
[HumanMessage(content='whats the weather in sf?', additional_kwargs=  
{}, response_metadata={}, id='15979f0b-4a79-46df-8d28-43bc289aa6'),  
AIMessage(content="I apologize, but I don't have a specific tool available  
to check the current weather in San Francisco. However, I can use the  
search function to find recent weather information for you. Would you like  
me to search for the current weather in San Francisco?",  
additional_kwargs={}, response_metadata={'ResponseMetadata':  
{'RequestId': 'c3d31db4-d840-49eb-8767-48eba5f89',  
'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Fri, 13 Dec 2024  
07:26:24 GMT', 'content-type': 'application/json', 'content-length': '439',  
'connection': 'keep-alive', 'x-amzn-requestid': 'c3d31db4-d840-49eb-
```

```
8767-48e9cb9'}, 'RetryAttempts': 0}, 'stopReason': 'end_turn', 'metrics':  
{'latencyMs': [1670]}}, id='run-d819b59d-2a1e-4d83-85a3-e714a5c2-0',  
usage_metadata={'input_tokens': 404, 'output_tokens': 55, 'total_tokens':  
459}}]
```

↑ 1 

2 replies



YanzhongSu [Dec 31, 2024](#)

How was your **agent_executor** initialised?



dantefu9001 [Jan 2](#)

had the same problem, need to use `print(response["message"])` to see the response



yanjiecdh [Dec 20, 2024](#)

hi, Can this method execute multiple tools consecutively, and can the execution results of the tools be passed on to the next tool

↑ 1 

1 reply



syntaxtrash [Feb 25](#)

tools are essentially functions wrapped in a specific format so the LLM can choose. You can totally do this by creating a custom tool that calls multiple tools in sequence and passes the output along.

Another way is to give the LLM clear instructions on how to use the tools in order, either in the prompt or tool descriptions.



dophat1 [Jan 9](#)

does anyone encounter this error while running til the define the tool step. I just copied all and paste it (i have set up my api key in langchain and tavily) and run the program in jupyter. It tells that:

"Failed to multipart ingest runs: langsmith.utils.LangSmithError: Failed to POST <https://eu.api.smith.langchain.com/runs/multipart> in LangSmith API. HTTPError('403 Client Error: Forbidden for url: <https://eu.api.smith.langchain.com/runs/multipart>', '{"detail": "Forbidden"}')"

just go all over the internet to find the solution but all of them cant fix my problem.

↑ 1



1 reply



rnsv [Mar 19](#)

disable if you don't have langsmith key

```
export LANGCHAIN_TRACING_V2=false
```



2980897531 [Jan 15](#)

I run astream events in python3.10 and it works

run .astream_events in python3.10 and it works.

↑ 1 

0 replies



BrazilForever11 [Jan 16](#)

Running the very first piece of code generates error:

```
TypeError: "Could not resolve authentication method. Expected either api_key or auth_token to be set. Or for one of the X-API-Key or Authorization headers to be explicitly omitted" During task with name 'agent' and id '48979dbb-5a2a-55ee-304f-61b221040872'
```

This probably authentication with sonnet?

```
ChatAnthropic(model_name="claude-3-sonnet-20240229") ?
```

↑ 1 

1 reply



fabandalm [Jan 16](#)

get anthropic api key on <https://console.anthropic.com/login> OR you can use openai

-

```
from langchain_openai import ChatOpenAI
```

```
model = ChatOpenAI(  
    model="gpt-4o",  
    temperature=0,  
    max_tokens=None,  
    timeout=None,  
    max_retries=2,
```

```
api_key=userdata.get("openai_api_key"),
```

```
# base_url="...",  
# organization="...",  
# other params...
```

```
).bind_tools(tools)
```

```
*
```



GoogTech [Feb 12](#)



Thanks a lot, bro! It's the best LangChain tutorial I have ever learned!!!

↑ 1



0 replies



AravindReddy123 [Feb 20](#)

Thanks a lot for this Article. I have a doubt here. So when we ask LLM What's the weather in SF?, here LLM is making that reasoning to call the search tool? and how does the LLM figure out the input structure for the tool? Was that sent in Prompt internally?

↑ 1



8 replies



[Show 3 previous replies](#)



AravindReddy123 [Feb 26](#)



Arvindreddy123 [Feb 20](#)

Thank you so much. That makes sense. Any document that says "How can I increase my tokens"? or do I need to purchase the subscription?



syntaxtrash [Feb 26](#)

happy to help! I'm assuming you're talking about the total tokens you can use over time with a model provider. You could try free alternatives first, like running open-source models on your own machine with [ollama](#) or check out the models from [google's ai studio](#) or [grog](#). If you need more, you might have to get a subscription from you current model provider.



shan-max [Mar 3](#)

<Thanks a lot for this Article. I have a doubt here. So when we ask LLM What's the weather in SF?, here LLM is making that reasoning to call the search tool? and how does the LLM figure out the input structure for the tool? Was that sent in Prompt internally?

\>



shan-max [Mar 3](#)

It's social media





shan-max [Mar 3](#)

Any one can ask help or comment as they can and they wish.This is to test you how active you are with my URL.I started it sharing my own github MR.Reddy(SC).



Weber12321 [Feb 27](#)

ModuleNotFoundError: No module named 'langgraph.prebuilt' in langgraph 0.3.0

↑ 1



3 replies



Yoana01 [Mar 3](#)

Hello, I have the same problem, did you fix it?



pengfan7758258 [Mar 7](#)

My langgraph version is 0.3.5, you can try



GoogTech [Mar 22](#)

[@Weber12321 @pengfan7758258](#)

The best answer is shown below. It worked for everybody!!!

```
pip install --upgrade --force-reinstall langgraph
```



shan-max [Mar 3](#)

This is not the thing , first you don't know how to read the code and you are Debugging me , the simple threads.

↑ 1



0 replies



shan-max [Mar 4](#)

Also, how can I rely on the third-party tool implementations, to give my users proper responses.

I cannot be testing for all possible inputs to these tools.

With above ambiguity, I am relegated to writing agents and my own custom tools

Tools appear to be wrappers over remote REST API, with a technical interface like runnable so that the agent can auto-invoke these tools

its a meek attempt to standardize AI agents with public REST APIs

↑ 1



0 replies



shan-max [Mar 4](#)

assuming there are multiple agents tools being bound into the model, how does "a qualitative input NLP question" trigger appropriate tool? there

... a question, answer question trigger appropriate tool which
could be multiple tools bound which are capable of providing the answer?



0 replies



shan-max [Mar 4](#)

<Thanks a lot for this Article. I have a doubt here. So when we ask LLM
What's the weather in SF?, here LLM is making that reasoning to call the
search tool? and how does the LLM figure out the input structure for the