This is documentation for LangChain **v0.1**, which is no longer actively maintained.

For the current stable version, see **this version** (Latest).

Quickstart

To best understand the agent framework, let's build an agent that has two tools: one to look things up online, and one to look up specific data that we've loaded into a index.

This will assume knowledge of LLMs and retrieval so if you haven't already explored those sections, it is recommended you do so.

Setup: LangSmith

By definition, agents take a self-determined, input-dependent sequence of steps before returning a user-facing output. This makes debugging these systems particularly tricky, and observability particularly important. LangSmith is especially useful for such cases.

When building with LangChain, all steps will automatically be traced in LangSmith. To set up LangSmith we just need set the following environment variables:

```
export LANGCHAIN_TRACING_V2="true"
export LANGCHAIN_API_KEY="<your-api-key>"
```

Define tools

We first need to create the tools we want to use. We will use two tools: Tavily (to search online) and then a retriever over a local index we will create

Tavily

We have a built-in tool in LangChain to easily use Tavily search engine as tool. Note that this requires an API key - they have a free tier, but if you don't have one or don't want to create one, you can always ignore this step.

Once you create your API key, you will need to export that as:

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

```
from langchain_community.tools.tavily_search import
TavilySearchResults
```

API Reference:

- TavilySearchResults

```
search = TavilySearchResults()
```

```
search.invoke("what is the weather in SF")
```

```
[{'url': 'https://www.weatherapi.com/',
  'content': "{'location': {'name': 'San Francisco', 'region':
'California', 'country': 'United States of America', 'lat': 37.78,
'lon': -122.42, 'tz_id': 'America/Los_Angeles', 'localtime_epoch':
1712847697, 'localtime': '2024-04-11 8:01'}, 'current':
{'last_updated_epoch': 1712847600, 'last_updated': '2024-04-11 08:00',
'temp_c': 11.1, 'temp_f': 52.0, 'is_day': 1, 'condition': {'text':
'Partly cloudy', 'icon':
'//cdn.weatherapi.com/weather/64x64/day/116.png', 'code': 1003},
'wind_mph': 2.2, 'wind_kph': 3.6, 'wind_degree': 10, 'wind_dir': 'N',
'pressure_mb': 1015.0, 'pressure_in': 29.98, 'precip_mm': 0.0,
'precip_in': 0.0, 'humidity': 97, 'cloud': 25, 'feelslike_c': 11.5,
'feelslike_f': 52.6, 'vis_km': 14.0, 'vis_miles': 8.0, 'uv': 4.0,
'gust_mph': 2.8, 'gust_kph': 4.4}}"},
 {'url': 'https://www.yahoo.com/news/april-11-2024-san-francisco-
122026435.html',
  'content': "2024 NBA Mock Draft 6.0: Projections for every pick
following March Madness With the NCAA tournament behind us, here's an
updated look at Yahoo Sports' first- and second-round projections for
the ..."},
 {'url': 'https://world-weather.info/forecast/usa/san_francisco/april-
2024/',
  'content': 'Extended weather forecast in San Francisco. Hourly Week
10 days 14 days 30 days Year. Detailed ⚡ San Francisco Weather Forecast
for April 2024 - day/night 🌡️ temperatures, precipitations - World-
Weather.info.'},
 {'url': 'https://www.wunderground.com/hourly/us/ca/san-
francisco/94144/date/date/2024-4-11',
  'content': 'Personal Weather Station. Inner Richmond (KCASANFR1685)
Location: San Francisco, CA. Elevation: 207ft. Nearby Weather Stations.
Hourly Forecast for Today, Thursday 04/11Hourly for Today, Thu 04/11
...'},
 {'url': 'https://weatherspark.com/h/y/557/2024/Historical-Weather-
during-2024-in-San-Francisco-California-United-States',
  'content': 'San Francisco Temperature History 2024\nHourly
Temperature in 2024 in San Francisco\nCompare San Francisco to another
city:\nCloud Cover in 2024 in San Francisco\nDaily Precipitation in
2024 in San Francisco\nObserved Weather in 2024 in San Francisco\nHours
of Daylight and Twilight in 2024 in San Francisco\nSunrise & Sunset
with Twilight and Daylight Saving Time in 2024 in San Francisco\nSolar
Elevation and Azimuth in 2024 in San Francisco\nMoon Rise, Set & Phases
in 2024 in San Francisco\nHumidity Comfort Levels in 2024 in San
Francisco\nWind Speed in 2024 in San Francisco\nHourly Wind Speed in
2024 in San Francisco\nHourly Wind Direction in 2024 in San
Francisco\nAtmospheric Pressure in 2024 in San Francisco\nData
Sources\n See all nearby weather stations\nLatest Report — 3:56
PM\nWed, Jan 24, 2024\xa0\xa0\xa0\xa013 min ago\xa0\xa0\xa0\xa0UTC
23:56\nCall Sign KSFO\nTemp.\n60.1°F\nPrecipitation\nNo
Report\nWind\n6.9 mph\nCloud Cover\nMostly Cloudy\n1,800 ft\nRaw: KSFO
242356Z 18006G19KT 10SM FEW015 BKN018 BKN039 16/12 A3004 RMK AO2 SLP171
T01560122 10156 20122 55001\n While having the tremendous advantages of
temporal and spatial completeness, these reconstructions: (1) are based
on computer models that may have model-based errors, (2) are coarsely
sampled on a 50 km grid and are therefore unable to reconstruct the
local variations of many microclimates, and (3) have particular
difficulty with the weather in some coastal areas, especially small
islands.\n We further caution that our travel scores are only as good
as the data that underpin them, that weather conditions at any given
location and time are unpredictable and variable, and that the
definition of the scores reflects a particular set of preferences that
may not agree with those of any particular reader.\n 2024 Weather
History in San Francisco California, United States\nThe data for this
report comes from the San Francisco International Airport.'}]
```

Retriever

We will also create a retriever over some data of our own. For a deeper explanation of each step here, see this section.

```python
from langchain_community.document_loaders import WebBaseLoader
from langchain_community.vectorstores import FAISS
from langchain_openai import OpenAIEmbeddings
from langchain_text_splitters import RecursiveCharacterTextSplitter

loader = WebBaseLoader("https://docs.smith.langchain.com/overview")
docs = loader.load()
documents = RecursiveCharacterTextSplitter(
    chunk_size=1000, chunk_overlap=200
).split_documents(docs)
vector = FAISS.from_documents(documents, OpenAIEmbeddings())
retriever = vector.as_retriever()
```

API Reference:

- WebBaseLoader
- FAISS
- OpenAIEmbeddings
- RecursiveCharacterTextSplitter

```python
retriever.invoke("how to upload a dataset")[0]
```

```
Document(page_content='import Clientfrom langsmith.evaluation import
evaluateclient = Client()# Define dataset: these are your test
casesdataset_name = "Sample Dataset"dataset =
client.create_dataset(dataset_name, description="A sample dataset in
LangSmith.")client.create_examples(    inputs=[        {"postfix": "to
LangSmith"},        {"postfix": "to Evaluations in LangSmith"},    ],
outputs=[        {"output": "Welcome to LangSmith"},        {"output":
"Welcome to Evaluations in LangSmith"},    ],
dataset_id=dataset.id,)# Define your evaluatordef exact_match(run,
example):    return {"score": run.outputs["output"] ==
example.outputs["output"]}experiment_results = evaluate(    lambda
input: "Welcome " + input[\'postfix\'], # Your AI system goes here
data=dataset_name, # The data to predict and grade over    evaluators=
[exact_match], # The evaluators to score the results
experiment_prefix="sample-experiment", # The name of the experiment
metadata={      "version": "1.0.0",      "revision_id":', metadata=
{'source': 'https://docs.smith.langchain.com/overview', 'title':
'Getting started with LangSmith | 🦜🛠️ LangSmith', 'description':
'Introduction', 'language': 'en'})
```

Now that we have populated our index that we will do doing retrieval over, we can easily turn it into a tool (the format needed for an agent to properly use it)

```
from langchain.tools.retriever import create_retriever_tool
```

API Reference:

- create_retriever_tool

```
retriever_tool = create_retriever_tool(
    retriever,
    "langsmith_search",
    "Search for information about LangSmith. For any questions about
LangSmith, you must use this tool!",
)
```

Tools

Now that we have created both, we can create a list of tools that we will use downstream.

```
tools = [search, retriever_tool]
```

Create the agent

Now that we have defined the tools, we can create the agent. We will be using an OpenAI Functions agent - for more information on this type of agent, as well as other options, see this guide.

First, we choose the LLM we want to be guiding the agent.

```
from langchain_openai import ChatOpenAI

llm = ChatOpenAI(model="gpt-3.5-turbo-0125", temperature=0)
```

API Reference:

- ChatOpenAI

Next, we choose the prompt we want to use to guide the agent.

If you want to see the contents of this prompt and have access to LangSmith, you can go to:

https://smith.langchain.com/hub/hwchase17/openai-functions-agent

```
from langchain import hub

# Get the prompt to use - you can modify this!
prompt = hub.pull("hwchase17/openai-functions-agent")
prompt.messages
```

```
[SystemMessagePromptTemplate(prompt=PromptTemplate(input_variables=[],
template='You are a helpful assistant')),
 MessagesPlaceholder(variable_name='chat_history', optional=True),
 HumanMessagePromptTemplate(prompt=PromptTemplate(input_variables=
['input'], template='{input}')),
 MessagesPlaceholder(variable_name='agent_scratchpad')]
```

Now, we can initialize the agent with the LLM, the prompt, and the tools. The agent is responsible for taking in input and deciding what actions to take. Crucially, the Agent does not execute those actions - that is done by the AgentExecutor (next step). For more information about how to think about these components, see our conceptual guide.

```
from langchain.agents import create_tool_calling_agent

agent = create_tool_calling_agent(llm, tools, prompt)
```

API Reference:

- create_tool_calling_agent

Finally, we combine the agent (the brains) with the tools inside the AgentExecutor (which will repeatedly call the agent and execute tools).

```
from langchain.agents import AgentExecutor

agent_executor = AgentExecutor(agent=agent, tools=tools, verbose=True)
```

API Reference:

- AgentExecutor

Run the agent

We can now run the agent on a few queries! Note that for now, these are all **stateless** queries (it won't remember previous interactions).

```
agent_executor.invoke({"input": "hi!"})
```

```


 [1m> Entering new AgentExecutor chain... [0m
 [32;1m [1;3mHello! How can I assist you today? [0m

 [1m> Finished chain. [0m
```

```
{'input': 'hi!', 'output': 'Hello! How can I assist you today?'}
```

```
agent_executor.invoke({"input": "how can langsmith help with
testing?"})
```

```


 [1m> Entering new AgentExecutor chain... [0m
 [32;1m [1;3m
Invoking: `langsmith_search` with `{'query': 'how can LangSmith help
with testing'}`


 [0m [33;1m [1;3mGetting started with LangSmith | 🦜🛠️ LangSmith

Skip to main contentLangSmith API DocsSearchGo to AppQuick StartUser
GuideTracingEvaluationProduction Monitoring & AutomationsPrompt
HubProxyPricingSelf-HostingCookbookQuick StartOn this pageGetting
started with LangSmithIntroductionLangSmith is a platform for building
production-grade LLM applications. It allows you to closely monitor and
evaluate your application, so you can ship quickly and with confidence.
Use of LangChain is not necessary - LangSmith works on its own!Install
```

```
LangSmithWe offer Python and Typescript SDKs for all your LangSmith
needs.PythonTypeScriptpip install -U langsmithyarn add langchain
langsmithCreate an API keyTo create an API key head to the setting
pages. Then click Create API Key.Setup your environmentShellexport
LANGCHAIN_TRACING_V2=trueexport LANGCHAIN_API_KEY=<your-api-key># The
below examples use the OpenAI API, though it's not necessary in
generalexport OPENAI_API_KEY=<your-openai-api-key>Log your first trace
We provide multiple ways to log traces

Learn about the workflows LangSmith supports at each stage of the LLM
application lifecycle.Pricing: Learn about the pricing model for
LangSmith.Self-Hosting: Learn about self-hosting options for
LangSmith.Proxy: Learn about the proxy capabilities of
LangSmith.Tracing: Learn about the tracing capabilities of
LangSmith.Evaluation: Learn about the evaluation capabilities of
LangSmith.Prompt Hub Learn about the Prompt Hub, a prompt management
tool built into LangSmith.Additional ResourcesLangSmith Cookbook: A
collection of tutorials and end-to-end walkthroughs using
LangSmith.LangChain Python: Docs for the Python LangChain
library.LangChain Python API Reference: documentation to review the
core APIs of LangChain.LangChain JS: Docs for the TypeScript LangChain
libraryDiscord: Join us on our Discord to discuss all things
LangChain!FAQHow do I migrate projects between organizations?Currently
we do not support project migration betwen organizations. While you can
manually imitate this by

team deals with sensitive data that cannot be logged. How can I ensure
that only my team can access it?If you are interested in a private
deployment of LangSmith or if you need to self-host, please reach out
to us at sales@langchain.dev. Self-hosting LangSmith requires an annual
enterprise license that also comes with support and formalized access
to the LangChain team.Was this page helpful?NextUser
GuideIntroductionInstall LangSmithCreate an API keySetup your
environmentLog your first traceCreate your first evaluationNext
StepsAdditional ResourcesFAQHow do I migrate projects between
organizations?Why aren't my runs aren't showing up in my project?My
team deals with sensitive data that cannot be logged. How can I ensure
that only my team can access it?CommunityDiscordTwitterGitHubDocs
CodeLangSmith SDKPythonJS/TSMoreHomepageBlogLangChain Python
DocsLangChain JS/TS DocsCopyright © 2024 LangChain,
Inc. [0m [32;1m [1;3mLangSmith is a platform for building production-
grade LLM applications that can help with testing in the following
ways:

1. **Tracing**: LangSmith provides tracing capabilities that allow you
to closely monitor and evaluate your application during testing. You
can log traces to track the behavior of your application and identify
```

```
any issues.

2. **Evaluation**: LangSmith offers evaluation capabilities that enable
you to assess the performance of your application during testing. This
helps you ensure that your application functions as expected and meets
the required standards.

3. **Production Monitoring & Automations**: LangSmith allows you to
monitor your application in production and automate certain processes,
which can be beneficial for testing different scenarios and ensuring
the stability of your application.

4. **Prompt Hub**: LangSmith includes a Prompt Hub, a prompt management
tool that can streamline the testing process by providing a centralized
location for managing prompts and inputs for your application.

Overall, LangSmith can assist with testing by providing tools for
monitoring, evaluating, and automating processes to ensure the
reliability and performance of your application during testing
phases. [0m

 [1m> Finished chain. [0m
```

```
{'input': 'how can langsmith help with testing?',
 'output': 'LangSmith is a platform for building production-grade LLM
applications that can help with testing in the following ways:\n\n1.
**Tracing**: LangSmith provides tracing capabilities that allow you to
closely monitor and evaluate your application during testing. You can
log traces to track the behavior of your application and identify any
issues.\n\n2. **Evaluation**: LangSmith offers evaluation capabilities
that enable you to assess the performance of your application during
testing. This helps you ensure that your application functions as
expected and meets the required standards.\n\n3. **Production
Monitoring & Automations**: LangSmith allows you to monitor your
application in production and automate certain processes, which can be
beneficial for testing different scenarios and ensuring the stability
of your application.\n\n4. **Prompt Hub**: LangSmith includes a Prompt
Hub, a prompt management tool that can streamline the testing process
by providing a centralized location for managing prompts and inputs for
your application.\n\nOverall, LangSmith can assist with testing by
providing tools for monitoring, evaluating, and automating processes to
ensure the reliability and performance of your application during
testing phases.'}
```

```python
agent_executor.invoke({"input": "whats the weather in sf?"})
```

```
 [1m> Entering new AgentExecutor chain... [0m
 [32;1m [1;3m
Invoking: `tavily_search_results_json` with `{'query': 'weather in San
Francisco'}`


 [0m [36;1m [1;3m[{'url': 'https://www.weatherapi.com/', 'content': "
{'location': {'name': 'San Francisco', 'region': 'California',
'country': 'United States of America', 'lat': 37.78, 'lon': -122.42,
'tz_id': 'America/Los_Angeles', 'localtime_epoch': 1712847697,
'localtime': '2024-04-11 8:01'}, 'current': {'last_updated_epoch':
1712847600, 'last_updated': '2024-04-11 08:00', 'temp_c': 11.1,
'temp_f': 52.0, 'is_day': 1, 'condition': {'text': 'Partly cloudy',
'icon': '//cdn.weatherapi.com/weather/64x64/day/116.png', 'code':
1003}, 'wind_mph': 2.2, 'wind_kph': 3.6, 'wind_degree': 10, 'wind_dir':
'N', 'pressure_mb': 1015.0, 'pressure_in': 29.98, 'precip_mm': 0.0,
'precip_in': 0.0, 'humidity': 97, 'cloud': 25, 'feelslike_c': 11.5,
'feelslike_f': 52.6, 'vis_km': 14.0, 'vis_miles': 8.0, 'uv': 4.0,
'gust_mph': 2.8, 'gust_kph': 4.4}}"}, {'url':
'https://www.yahoo.com/news/april-11-2024-san-francisco-
122026435.html', 'content': "2024 NBA Mock Draft 6.0: Projections for
every pick following March Madness With the NCAA tournament behind us,
here's an updated look at Yahoo Sports' first- and second-round
projections for the ..."}, {'url':
'https://www.weathertab.com/en/c/e/04/united-states/california/san-
francisco/', 'content': 'Explore comprehensive April 2024 weather
forecasts for San Francisco, including daily high and low temperatures,
precipitation risks, and monthly temperature trends. Featuring detailed
day-by-day forecasts, dynamic graphs of daily rain probabilities, and
temperature trends to help you plan ahead. ... 11 65°F 49°F 18°C 9°C
29% 12 64°F 49°F ...'}, {'url':
'https://weatherspark.com/h/y/557/2024/Historical-Weather-during-2024-
in-San-Francisco-California-United-States', 'content': 'San Francisco
Temperature History 2024\nHourly Temperature in 2024 in San
Francisco\nCompare San Francisco to another city:\nCloud Cover in 2024
in San Francisco\nDaily Precipitation in 2024 in San
Francisco\nObserved Weather in 2024 in San Francisco\nHours of Daylight
and Twilight in 2024 in San Francisco\nSunrise & Sunset with Twilight
and Daylight Saving Time in 2024 in San Francisco\nSolar Elevation and
Azimuth in 2024 in San Francisco\nMoon Rise, Set & Phases in 2024 in
San Francisco\nHumidity Comfort Levels in 2024 in San Francisco\nWind
```

```
Speed in 2024 in San Francisco\nHourly Wind Speed in 2024 in San
Francisco\nHourly Wind Direction in 2024 in San Francisco\nAtmospheric
Pressure in 2024 in San Francisco\nData Sources\n See all nearby
weather stations\nLatest Report — 3:56 PM\nWed, Jan 24,
2024\xa0\xa0\xa0\xa013 min ago\xa0\xa0\xa0\xa0UTC 23:56\nCall Sign
KSFO\nTemp.\n60.1°F\nPrecipitation\nNo Report\nWind\n6.9 mph\nCloud
Cover\nMostly Cloudy\n1,800 ft\nRaw: KSFO 242356Z 18006G19KT 10SM
FEW015 BKN018 BKN039 16/12 A3004 RMK AO2 SLP171 T01560122 10156 20122
55001\n While having the tremendous advantages of temporal and spatial
completeness, these reconstructions: (1) are based on computer models
that may have model-based errors, (2) are coarsely sampled on a 50 km
grid and are therefore unable to reconstruct the local variations of
many microclimates, and (3) have particular difficulty with the weather
in some coastal areas, especially small islands.\n We further caution
that our travel scores are only as good as the data that underpin them,
that weather conditions at any given location and time are
unpredictable and variable, and that the definition of the scores
reflects a particular set of preferences that may not agree with those
of any particular reader.\n 2024 Weather History in San Francisco
California, United States\nThe data for this report comes from the San
Francisco International Airport.'}, {'url': 'https://www.msn.com/en-
us/weather/topstories/april-11-2024-san-francisco-bay-area-weather-
forecast/vi-BB1lrXDb', 'content': 'April 11, 2024 San Francisco Bay
Area weather forecast. Posted: April 11, 2024 | Last updated: April 11,
2024 ...'}] [0m [32;1m [1;3mThe current weather in San Francisco is
partly cloudy with a temperature of 52.0°F (11.1°C). The wind speed is
3.6 kph coming from the north, and the humidity is at 97%. [0m

 [1m> Finished chain. [0m
```

```
{'input': 'whats the weather in sf?',
 'output': 'The current weather in San Francisco is partly cloudy with
a temperature of 52.0°F (11.1°C). The wind speed is 3.6 kph coming from
the north, and the humidity is at 97%.'}
```

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 previous `chat_history`. Note: it needs to be called `chat_history` because of the prompt we are using. If we use a different prompt, we could change the variable name

```
# Here we pass in an empty list of messages for chat_history because
it is the first message in the chat
agent_executor.invoke({"input": "hi! my name is bob", "chat_history":
[]})
```

```


[1m> Entering new AgentExecutor chain... [0m
[32;1m [1;3mHello Bob! How can I assist you today? [0m

[1m> Finished chain. [0m
```

```
{'input': 'hi! my name is bob',
 'chat_history': [],
 'output': 'Hello Bob! How can I assist you today?'}
```

```
from langchain_core.messages import AIMessage, HumanMessage
```

API Reference:

- AIMessage
- HumanMessage

```
agent_executor.invoke(
    {
        "chat_history": [
            HumanMessage(content="hi! my name is bob"),
            AIMessage(content="Hello Bob! How can I assist you
today?"),
        ],
        "input": "what's my name?",
    }
)
```

```


[1m> Entering new AgentExecutor chain... [0m
[32;1m [1;3mYour name is Bob. How can I assist you, Bob? [0m
```

```
[1m> Finished chain. [0m
```

```
{'chat_history': [HumanMessage(content='hi! my name is bob'),
  AIMessage(content='Hello Bob! How can I assist you today?')],
 'input': "what's my name?",
 'output': 'Your name is Bob. How can I assist you, Bob?'}
```

If we want to keep track of these messages automatically, we can wrap this in a RunnableWithMessageHistory. For more information on how to use this, see this guide.

```
from langchain_community.chat_message_histories import
ChatMessageHistory
from langchain_core.runnables.history import
RunnableWithMessageHistory
```

API Reference:

- ChatMessageHistory
- RunnableWithMessageHistory

```
message_history = ChatMessageHistory()
```

```
agent_with_chat_history = RunnableWithMessageHistory(
    agent_executor,
    # This is needed because in most real world scenarios, a session
id is needed
    # It isn't really used here because we are using a simple in
memory ChatMessageHistory
    lambda session_id: message_history,
    input_messages_key="input",
    history_messages_key="chat_history",
)
```

```
agent_with_chat_history.invoke(
    {"input": "hi! I'm bob"},
    # This is needed because in most real world scenarios, a session
id is needed
    # It isn't really used here because we are using a simple in
memory ChatMessageHistory
    config={"configurable": {"session_id": "<foo>"}},
)
```

```
[1m> Entering new AgentExecutor chain... [0m
[32;1m [1;3mHello Bob! How can I assist you today? [0m

[1m> Finished chain. [0m
```

```
{'input': "hi! I'm bob",
 'chat_history': [],
 'output': 'Hello Bob! How can I assist you today?'}
```

```
agent_with_chat_history.invoke(
    {"input": "what's my name?"},
    # This is needed because in most real world scenarios, a session
id is needed
    # It isn't really used here because we are using a simple in
memory ChatMessageHistory
    config={"configurable": {"session_id": "<foo>"}},
)
```

```
[1m> Entering new AgentExecutor chain... [0m
[32;1m [1;3mYour name is Bob! How can I help you, Bob? [0m

[1m> Finished chain. [0m
```

```
{'input': "what's my name?",
 'chat_history': [HumanMessage(content="hi! I'm bob"),
  AIMessage(content='Hello Bob! How can I assist you today?')],
 'output': 'Your name is Bob! How can I help you, Bob?'}
```

Conclusion

That's a wrap! In this quick start we covered how to create a simple agent. Agents are a complex topic, and there's lot to learn! Head back to the main agent page to find more resources on conceptual guides, different types of agents, how to create custom tools, and more!

Help us out by providing feedback on this documentation page: