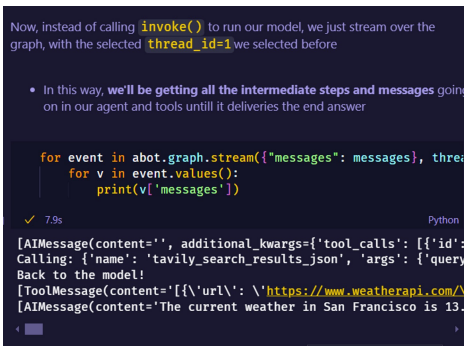
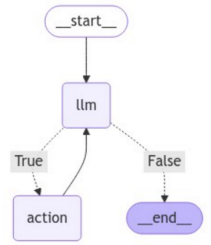
Streaming and Persistence

viernes, noviembre 01, 2024 1:53 PM



1. Adding persistence to the model.
2. Adding streaming
   a. **Both can be synchronous**
   b. **Or asynchronous**



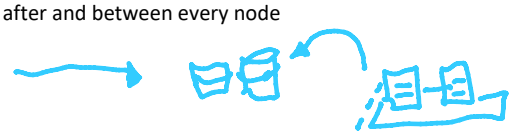
Appendix (in notebooks) **HUMAN IN THE LOOP INTERACTIONS**

We can add interrumpts in different nodes like node="action" so whenever The agent needs to run a node (e.g action), it will stop there, and from there We decide whether resuming the graph or stopping the agent

1. Persistence

The **persistence** allows us to checkpoint
The state of the agent after and between every node

a. It can be synchronous
b. Or asynchronous



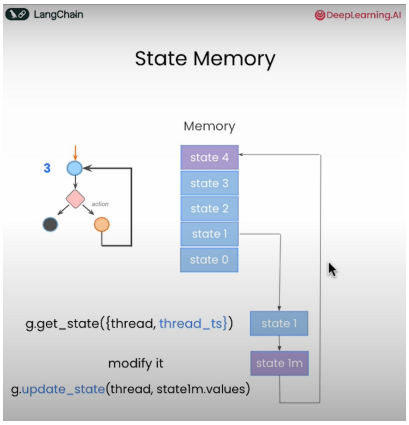
Waits for the llm/tool
To finish in order to checkpoint it.
(stores everything sequentially)

Stores the
checkpoints
In the database
without stopping
Any process

```
thread = {"configurable": {"thread_id": "1"}}
```

We are using threads.
This allows us in real world apps to
Have multiple conversations in pararell

- **Whereas the agents, can renmber from the Context.**
  E.g if you use a different thread_id, it won't renember anything



Extra note: We can acess to all the different "states" inside the llm,
And changue them

We can also "Mock out" responses
For the llm instead of calling an
Action node, so it directly goes to the
Llm...etc

(details in notebooks)

2. Streaming

```
abot.graph.stream({"messages": messages}, thread):
```

Depending on the persistence configuration (agent graph memmory saver) we selected: **we'll see the intermediate steps of the agent and output**

A) It can be **synchronous**

```
{'messages': [AIMessage(content='', additional_kwargs={'tool_calls': [{'id':
Calling: {'name': 'tavily_search_results_json', 'args': {'query': 'Los Angel
Back to the model!
{'messages': [ToolMessage(content='[{\'url\': \'https://www.weatherapi.com/\
{'messages': [AIMessage(content='The current weather in Los Angeles is 15.1°
```

B) Or it can be **asynchronous**

The LLM now streams tokens too when running,
So we can see each generated **token** in real time

```
Calling: {'name': 'tavily_search_results_json', 'args': {'query': 'current p
Calling: {'name': 'tavily_search_results_json', 'args': {'query': 'current p
Back to the model!
In| San| Francisco|,| the| average| price| for| |1| kilogram| (|approximatel

|In| Spain|,| specifically| in| Barcelona|,| the| price| for| |1| kilogram|

|Compar|atively|,| tomatoes| are| significantly| more| expensive| in| San| F
```

(this | was the separator
Between tokens)