



Streaming final agent output

If you only want the final output of an agent to be streamed, you can use the callback `FinalStreamingStdOutCallbackHandler`. For this, the underlying LLM has to support streaming as well.

```
from langchain.agents import load_tools
from langchain.agents import initialize_agent
from langchain.agents import AgentType
from langchain.callbacks.streaming_stdout_final_only import (
    FinalStreamingStdOutCallbackHandler,
)
from langchain.llms import OpenAI
```

API Reference:

- `load_tools` from `langchain.agents`
- `initialize_agent` from `langchain.agents`
- `AgentType` from `langchain.agents`
- `FinalStreamingStdOutCallbackHandler` from `langchain.callbacks.streaming_stdout_final_only`
- `OpenAI` from `langchain.llms`

Let's create the underlying LLM with `streaming = True` and pass a new instance of `FinalStreamingStdOutCallbackHandler`.

```
llm = OpenAI(
    streaming=True, callbacks=[FinalStreamingStdOutCallbackHandler()],
    temperature=0
)
```

```
tools = load_tools(["wikipedia", "llm-math"], llm=llm)
agent = initialize_agent(
    tools, llm, agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION, verbose=False
```



```
)
agent.run(
    "It's 2023 now. How many years ago did Konrad Adenauer become
    Chancellor of Germany."
)
```

Konrad Adenauer became Chancellor of Germany in 1949, 74 years ago in 2023.

'Konrad Adenauer became Chancellor of Germany in 1949, 74 years ago in 2023.'

Handling custom answer prefixes

By default, we assume that the token sequence `"Final", "Answer", ":"` indicates that the agent has reached an answers. We can, however, also pass a custom sequence to use as answer prefix.

```
llm = OpenAI(
    streaming=True,
    callbacks=[
        FinalStreamingStdOutCallbackHandler(answer_prefix_tokens=["The",
"answer", ":"])
    ],
    temperature=0,
)
```

For convenience, the callback automatically strips whitespaces and new line characters when comparing to `answer_prefix_tokens`. I.e., if `answer_prefix_tokens = ["The", "answer", ":"]` then both `["\nThe", " answer", ":"]` and `["The", " answer", ":"]` would be recognized as the answer prefix.

If you don't know the tokenized version of your answer prefix, you can determine it with the following code:

```

from langchain.callbacks.base import BaseCallbackHandler

class MyCallbackHandler(BaseCallbackHandler):
    def on_llm_new_token(self, token, **kwargs) -> None:
        # print every token on a new line
        print(f"#{token}#")

llm = OpenAI(streaming=True, callbacks=[MyCallbackHandler()])
tools = load_tools(["wikipedia", "llm-math"], llm=llm)
agent = initialize_agent(
    tools, llm, agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION, verbose=False
)
agent.run(
    "It's 2023 now. How many years ago did Konrad Adenauer become
    Chancellor of Germany."
)

```

API Reference:

- `BaseCallbackHandler` from `langchain.callbacks.base`

Also streaming the answer prefixes

When the parameter `stream_prefix = True` is set, the answer prefix itself will also be streamed. This can be useful when the answer prefix itself is part of the answer. For example, when your answer is a JSON like

```
{ "action": "Final answer", "action_input": "Konrad Adenauer became
Chancellor 74 years ago." }
```

and you don't only want the `action_input` to be streamed, but the entire JSON.