



Custom callback handlers

You can create a custom handler to set on the object as well. In the example below, we'll implement streaming with a custom handler.

```
from langchain.callbacks.base import BaseCallbackHandler
from langchain.chat_models import ChatOpenAI
from langchain.schema import HumanMessage

class MyCustomHandler(BaseCallbackHandler):
    def on_llm_new_token(self, token: str, **kwargs) -> None:
        print(f"My custom handler, token: {token}")

# To enable streaming, we pass in `streaming=True` to the ChatModel
# constructor
# Additionally, we pass in a list with our custom handler
chat = ChatOpenAI(max_tokens=25, streaming=True, callbacks=
[MyCustomHandler()])

chat([HumanMessage(content="Tell me a joke")])
```

API Reference:

- [BaseCallbackHandler](#) from `langchain.callbacks.base`
- [ChatOpenAI](#) from `langchain.chat_models`
- [HumanMessage](#) from `langchain.schema`

```
My custom handler, token:
My custom handler, token: Why
My custom handler, token: don
My custom handler, token: 't
My custom handler, token: scientists
My custom handler, token: trust
My custom handler, token: atoms
My custom handler, token: ?
My custom handler, token:
```

```
My custom handler, token: Because
My custom handler, token:  they
My custom handler, token:  make
My custom handler, token:  up
My custom handler, token:  everything
My custom handler, token:  .
My custom handler, token:
```

```
AIMessage(content="Why don't scientists trust atoms? \n\nBecause
they make up everything.", additional_kwargs={}, example=False)
```