



Streaming

Some Chat models provide a streaming response. This means that instead of waiting for the entire response to be returned, you can start processing it as soon as it's available. This is useful if you want to display the response to the user as it's being generated, or if you want to process the response as it's being generated.

```
from langchain.chat_models import ChatOpenAI
from langchain.schema import (
    HumanMessage,
)

from langchain.callbacks.streaming_stdout import StreamingStdOutCallbackHandler
chat = ChatOpenAI(streaming=True, callbacks=
[StreamingStdOutCallbackHandler()], temperature=0)
resp = chat([HumanMessage(content="Write me a song about sparkling
water.")])
```

Verse 1:

Bubbles rising to the top
A refreshing drink that never stops
Clear and crisp, it's pure delight
A taste that's sure to excite

Chorus:

Sparkling water, oh so fine
A drink that's always on my mind
With every sip, I feel alive
Sparkling water, you're my vibe

Verse 2:

No sugar, no calories, just pure bliss
A drink that's hard to resist
It's the perfect way to quench my thirst
A drink that always comes first

Chorus:

Sparkling water, oh so fine
A drink that's always on my mind



With every sip, I feel alive
Sparkling water, you're my vibe

Bridge:

From the mountains to the sea
Sparkling water, you're the key
To a healthy life, a happy soul
A drink that makes me feel whole

Chorus:

Sparkling water, oh so fine
A drink that's always on my mind
With every sip, I feel alive
Sparkling water, you're my vibe

Outro:

Sparkling water, you're the one
A drink that's always so much fun
I'll never let you go, my friend
Sparkling