



# Token counting

LangChain offers a context manager that allows you to count tokens.

```
import asyncio

from langchain.callbacks import get_openai_callback
from langchain.llms import OpenAI

llm = OpenAI(temperature=0)
with get_openai_callback() as cb:
    llm("What is the square root of 4?")

total_tokens = cb.total_tokens
assert total_tokens > 0

with get_openai_callback() as cb:
    llm("What is the square root of 4?")
    llm("What is the square root of 4?")

assert cb.total_tokens == total_tokens * 2

# You can kick off concurrent runs from within the context manager
with get_openai_callback() as cb:
    await asyncio.gather(
        *[llm.agenerate(["What is the square root of 4?"]) for _ in
range(3)]
    )

assert cb.total_tokens == total_tokens * 3

# The context manager is concurrency safe
task = asyncio.create_task(llm.agenerate(["What is the square root of
4?"]))
with get_openai_callback() as cb:
    await llm.agenerate(["What is the square root of 4?"])

await task
assert cb.total_tokens == total_tokens
```

## API Reference:

- `get_openai_callback` from `langchain.callbacks`

- **OpenAI** from `langchain.llms`