Documentation

Prefilling

Assistant Message Prefilling

When using Groq API, you can have more control over your model output by prefilling `assistant` messages. This technique gives you the ability to direct any text-to-text model powered by Groq to:

- Skip unnecessary introductions or preambles
- Enforce specific output formats (e.g., JSON, XML)
- Maintain consistency in conversations

How to Prefill Assistant messages

To prefill, simply include your desired starting text in the `assistant` message and the model will generate a response starting with the `assistant` message.

Note: For some models, adding a newline after the prefill `assistant` message leads to better results.

💡 **Tip:** Use the stop sequence (`stop`) parameter in combination with prefilling for even more concise results. We recommend using this for generating code snippets.

Examples

Example 1: Controlling output format for concise code snippets

When trying the below code, first try a request without the prefill and then follow up by trying another request with the prefill included to see the difference!

curl JavaScript Python JSON

````python
from groq import Groq

client = Groq()
completion = client.chat.completions.create(
    model="llama3-70b-8192",
    messages=[
        {
            "role": "user",
            "content": "Write a Python function to calculate the factorial of a number."
        },
        {
            "role": "assistant",
            "content": "```python"
        }
    ],
    stop="```",
)

for chunk in completion:
    print(chunk.choices[0].delta.content or "", end="")
````

Example 2: Extracting structured data from unstructured input

curl JavaScript Python JSON

```python
from groq import Groq

client = Groq()
completion = client.chat.completions.create(
```

```
    model="llama3-70b-8192",
    messages=[
        {
            "role": "user",
            "content": "Extract the title, author, published date, and description from the following book as a JS
        },
        {
            "role": "assistant",
            "content": "```json"
        }
    ],
    stop="```",
)

for chunk in completion:
    print(chunk.choices[0].delta.content or "", end="")
```