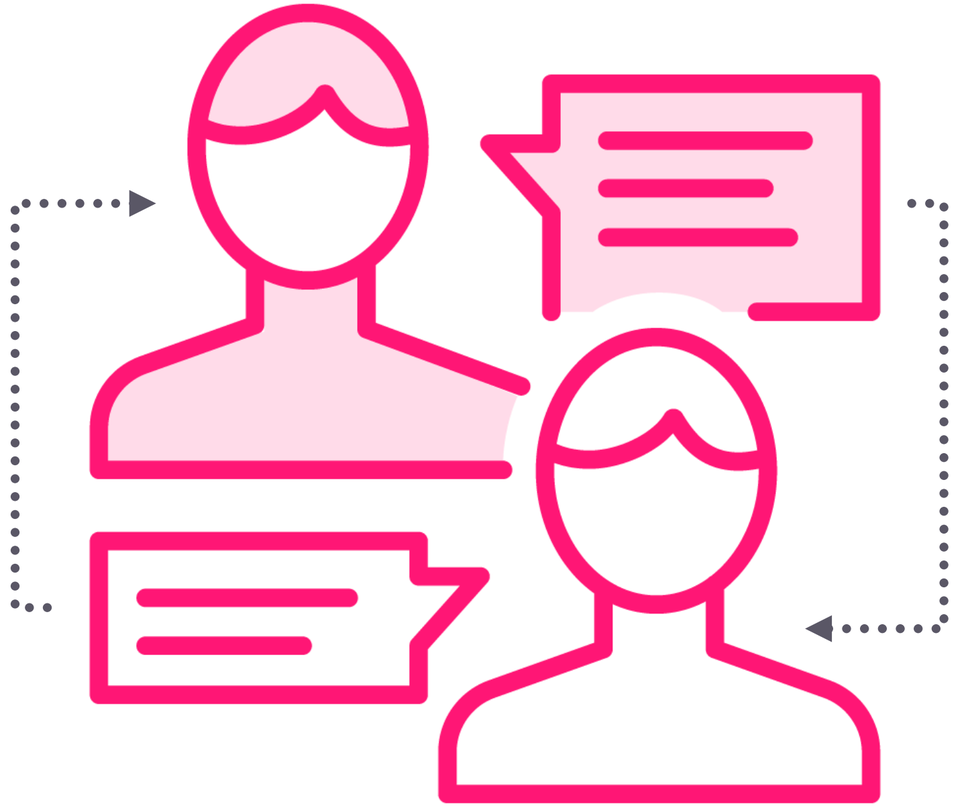
Using the Chat Completions API

Esteban Herrera

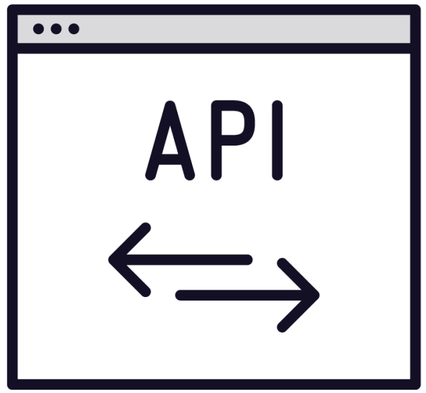
Author

@eh3rrera | eherrera.net

Multi-step Prompt



Demo Application



A command-line application

- It analyzes job interview transcriptions
- It reads the transcriptions from a TXT file
- First prompt extracts information from the interview, storing it in a JSON object
- Second prompt analyzes the information
- Function calling
  - Schedule a follow-up call

Designing the First Prompt

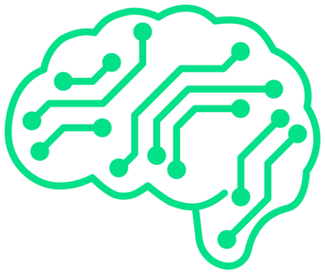
Designing the Second Prompt

Building the application

Calling External Functions

Function Calling

GPT model




```
{
  "role": "assistant",
  "tool_calls": [
    {
      "id": "call_id",
      "type": "function",
      "function": {
        "name": "send_email",
        "arguments": {
          "to": "john@example.com",
          "body": "Hi!”
        }
      }
    }
  ]
}
```

Step 1. Describe the Function

```
[
    {
        "type": "function",
        "function": {
            "name": "send_email",
            "description": "Sends an email",
            "parameters": {
                "type": "object",
                "properties": {
                    "to": {
                        "type": "string",
                        "description": "The email address of the recipient.",
                    },
                    "body": {
                        "type": "string",
                        "description": "The content or message to be sent in the email.",
                    },
                },
                "required": ["to", "body"],
            },
        }
    }
]
```

Step 2. The Model Generates a JSON Object to Call the Function

Send an email to John to say hi

```
{
  "role": "assistant",
  "tool_calls": [
    {
      "id": "call_id",
      "type": "function",
      "function": {
        "name": "send_email",
        "arguments": {
          "to": "john@example.com",
          "body": "Hi!”
        }
      }
    }
  ]
}
```

Step 3. (You) Call the Function

```
{
  "role": "assistant",
  "tool_calls": [
    {
      "id": "call_id",
      "type": "function",
      "function": {
        "name": "send_email",
        "arguments": {
          "to": "john@example.com",
          "body": "Hi!”
        }
      }
    }
  ]
}
```

```
send_email(
  to = "john@example.com",
  body = "Hi!"
)
```

Step 4. Send the Model the Result of the Function

```
{
  "tool_call_id": "call_id",
  "role": "tool",
  "name": "send_email",
  "arguments": "Email sent successfully"
}
```

Error Handling

Summary

The Chat Completions API demo

- Refine a multi-step prompt using the Playground
- First prompt extracts relevant information
- Second prompt analyzes the information
- Integrating the API code
- Using function calling
- Adding error handling

Up Next:

Optimizing API Usage