

ChatGpt API 활용

Day6. 구글 캘린더 API, Function Calling

구글 캘린더 API 활용

구글 캘린더 API 활용

채팅 서비스에 @calendar 추가해보기

- 사용자가 자연어로 입력하는 텍스트로 어떻게 API를 호출할까?
 - “내일 오전 9시 30분 부터 GPT활용 강의 캘린더에 추가해줘”
 - 구글 캘린더 API는 다음과 같은 JSON 포맷으로 호출해야함.

```
event = {  
    'summary': 'Lunch with my friends',  
    'start': {  
        'dateTime': start_time.isoformat(),  
        'timeZone': 'Asia/Seoul'  
    },  
    'end': {  
        'dateTime': end_time.isoformat(),  
        'timeZone': 'Asia/Seoul'  
    }  
}
```

구글 캘린더 API 활용

채팅 서비스에 @calendar 추가해보기

- 사용자가 자연어로 입력하는 텍스트로 어떻게 API를 호출할까?
 - “내일 오전 9시 30분 부터 GPT활용 강의 캘린더에 추가해줘”
 - 구글 캘린더 API는 다음과 같은 JSON 포맷으로 호출해야함.
- => GPT를 활용하여 자연어로 부터 JSON 형태의 정형화된 데이터를 추출하여 보자.

구글 캘린더 API 활용

채팅 서비스에 @calendar 추가해보기

- 어떤식으로 프롬프트를 주어야 할까요?

```
time = datetime.now(kst).isoformat()
system_msg = {
    "role": "system",
    "content": (
        f'''당신은 사용자의 일정 요청을 분석해 JSON 형식으로 응답하는 일정 파서입니다.
        기준이 되는 현재 시간은 다음과 같습니다 {time}
        형식은 다음과 같습니다:
        {{
            "action": "create" | "update" | "delete",
            "title": "지나랑 저녁 약속",
            "datetime": "2024-07-24T19:00:00",
            "new_datetime": "2024-07-24T20:00:00"
        }}
        오직 JSON으로만 응답해줘.'''
    )
}
```

구글 캘린더 API 활용

채팅 서비스에 @calendar 추가해보기

1. 프론트엔드로부터 메시지를 받으면 GPT API를 통해 JSON 형식으로 변환
 - @calendar 내일 8시 친구랑 약속
2. create 인 경우 일정 추가
 - `service.events().insert(calendarId=calendar_id, body=event).execute()`
3. delete, update 인 경우 기존 일정 찾고, 추가 또는 삭제
 - delete : `service.events().delete(calendarId=calendar_id, eventId=event["id"]).execute()`
 - update : `service.events().update(calendarId=calendar_id, eventId=event["id"], body=event).execute()`

구글 캘린더 API 활용

채팅 서비스에 @calendar 추가해보기

Chat

t:
@calendar 내일 지나랑 점심식사 12시
t:
시
calendar:
지나랑 점심식사 일정이 등록되었습니다. (2025-08-06T12:00:00)

파일 선택 선택된 파일 없음

Send Image

Start STT

Start Talk

Stop & Send

Function Calling

Function Calling

Function Calling이란?

- LLM 모델이 자연어 요청을 이해하고
 - 적절한 함수(Function)를 선택하고
 - 그 인자(Parameter)를 생성하여 호출하도록 설계된 매커니즘
-
- 모델이 단순히 텍스트를 생성하는 것을 넘어, 실제 코드 실행과 연동될 수 있도록 해줌

Function Calling

Function Calling 작동 예시

1. 사용자 입력 : 사용자가 자연어로 요청을 보냄
 - “내일 대전 날씨 알려줘”

Function Calling

Function Calling 작동 예시

2. 모델은 이 요청이 어떤 함수와 매핑 되는지 판단

- `get_weather(city="대전", date="2025-08-06")`

Function Calling

Function Calling 작동 예시

3. 모델은 함수 이름과 인자를 JSON 형식을 생성

```
{  
  "name": "get_weather",  
  "arguments" {  
    "city": "대전",  
    "date": "2025-08-07"  
  }  
}
```

Function Calling

Function Calling 작동 예시

4. 서버가 해당 함수를 실행

- JSON 정보를 바탕으로 실제 함수를 실행

Function Calling

Function Calling이란?

- Function Calling의 핵심 요소
 - 함수 목록
 - 모델에게 어떤 함수가 있고, 어떤 인자를 받는지 알려주는 스펙
 - 함수 선택
 - 모델이 어떤 함수를 호출할지 선택하는 방식
 - 함수 인자 추론
 - 사용자의 자연어 입력에서 인자를 추출하여 JSON 형태로 생성
 - 함수 실행
 - 사용자가 만든 서버가 이 JSON을 실행 가능한 코드로 처리

Function Calling

chat/completions API

- chat/completions API의 tools 파라미터에 함수 목록을 명시하여 Function Calling 기능 사용 가능

tools array Optional

A list of tools the model may call. Currently, only functions are supported as a tool. Use this to provide a list of functions the model may generate JSON inputs for. A max of 128 functions are supported.

^ Hide properties

function object **Required**

^ Hide properties

name string **Required**

The name of the function to be called. Must be a-z, A-Z, 0-9, or contain underscores and dashes, with a maximum length of 64.

description string Optional

A description of what the function does, used by the model to choose when and how to call the function.

parameters object Optional

The parameters the functions accepts, described as a JSON Schema object. See the [guide](#) for examples, and the [JSON Schema reference](#) for documentation about the format.

Omitting `parameters` defines a function with an empty parameter list.

strict boolean or null Optional Defaults to false

Whether to enable strict schema adherence when generating the function call. If set to true, the model will follow the exact schema defined in the `parameters` field. Only a subset of JSON Schema is supported when `strict` is `true`. Learn more about Structured Outputs in the [function calling guide](#).

type string **Required**

The type of the tool. Currently, only `function` is supported.

<https://platform.openai.com/docs/api-reference/chat/create>

Function Calling

함수 목록

- function
 - 실제 호출할 함수에 대한 정보를 담은 객체
- name
 - 함수 이름
- description
 - 함수가 하는 일을 설명. 모델이 이해하는데 도움
- parameter
 - 함수가 어떤 인자를 받는지를 정의

```
tools = [  
  {  
    "type": "function",  
    "function": {  
      "name": "say_hello",  
      "description": "사 용 자 에 게 인 사 합 니 다 .",  
      "parameters": {  
        "type": "object",  
        "properties": {  
          "name": { "type": "string" }  
        },  
        "required": ["name"]  
      }  
    }  
  },  
  {  
    "type": "function",  
    "function": {  
      "name": "add_numbers",  
      "description": "두 숫 자 를 더 합 니 다 .",  
      "parameters": {  
        "type": "object",  
        "properties": {  
          "a": { "type": "number" },  
          "b": { "type": "number" }  
        },  
        "required": ["a", "b"]  
      }  
    }  
  }  
]
```


Function Calling

함수 목록

- properties
 - 함수가 받을 수 있는 각 인자의 목록과 타입을 정의
- required
 - 반드시 있어야 하는 인자 이름을 배열로 명시

```
tools = [  
  {  
    "type": "function",  
    "function": {  
      "name": "say_hello",  
      "description": "사 용 자 에 게 인 사 합 니 다 .",  
      "parameters": {  
        "type": "object",  
        "properties": {  
          "name": { "type": "string" }  
        },  
        "required": ["name"]  
      }  
    },  
  },  
  {  
    "type": "function",  
    "function": {  
      "name": "add_numbers",  
      "description": "두 숫 자 를 더 합 니 다 .",  
      "parameters": {  
        "type": "object",  
        "properties": {  
          "a": { "type": "number" },  
          "b": { "type": "number" }  
        },  
        "required": ["a", "b"]  
      }  
    },  
  }  
]
```

Function Calling

함수 목록

- 사용자가 “지피티야 안녕!”을 입력했을 때

```
{  
  "name": "say_hello",  
  "arguments": {  
    "name": "수빈"  
  }  
}
```

```
tools = [  
  {  
    "type": "function",  
    "function": {  
      "name": "say_hello",  
      "description": "사용자에게 인사합니다.",  
      "parameters": {  
        "type": "object",  
        "properties": {  
          "name": { "type": "string" }  
        },  
        "required": ["name"]  
      }  
    }  
  },  
  {  
    "type": "function",  
    "function": {  
      "name": "add_numbers",  
      "description": "두 숫자를 더합니다.",  
      "parameters": {  
        "type": "object",  
        "properties": {  
          "a": { "type": "number" },  
          "b": { "type": "number" }  
        },  
        "required": ["a", "b"]  
      }  
    }  
  }  
]
```

Function Calling

함수 목록

- 사용자가 “2+7=?”을 입력했을 때

```
{
  "name": "add_numbers",
  "arguments": {
    "a": 3,
    "b": 7
  }
}
```

```
tools = [
  {
    "type": "function",
    "function": {
      "name": "say_hello",
      "description": "사 용 자 에 게 인 사 합 니 다 .",
      "parameters": {
        "type": "object",
        "properties": {
          "name": { "type": "string" }
        },
        "required": ["name"]
      }
    }
  },
  {
    "type": "function",
    "function": {
      "name": "add_numbers",
      "description": "두 숫 자 를 더 합 니 다 .",
      "parameters": {
        "type": "object",
        "properties": {
          "a": { "type": "number" },
          "b": { "type": "number" }
        },
        "required": ["a", "b"]
      }
    }
  }
]
```

Function Calling

chat/completions API

tool_choice string or object Optional

Controls which (if any) tool is called by the model. `none` means the model will not call any tool and instead generates a message. `auto` means the model can pick between generating a message or calling one or more tools. `required` means the model must call one or more tools. Specifying a particular tool via `{"type": "function", "function": {"name": "my_function"}}` forces the model to call that tool.

`none` is the default when no tools are present. `auto` is the default if tools are present.

▽ Show possible types

- tool_choice 파라미터는 모델이 어떤 함수를 호출할지 선택하는 방식을 제어
 - none : 함수를 호출하지 않고, 일반 텍스트 메시지만 생성
 - auto : 필요하면 도구를 호출, 아니면 일반 텍스트 응답
 - required : 모델이 반드시 도구 중 하나 이상을 호출
 - { “type” ~ } : 모델이 무조건 해당 이름의 함수 호출

Function Calling

실습

```
1 import json
2 from openai import OpenAI
3
4 client = OpenAI(api_key="YOUR_API_KEY")
5
6 tools = [
7     {
8         "type": "function",
9         "function": {
10             "name": "say_hello",
11             "description": "사용자에게 인사합니다.",
12             "parameters": {
13                 "type": "object",
14                 "properties": {
15                     "name": { "type": "string" }
16                 },
17                 "required": ["name"]
18             }
19         }
20     },
21     {
22         "type": "function",
23         "function": {
24             "name": "add_numbers",
25             "description": "두 숫자를 더합니다.",
26             "parameters": {
27                 "type": "object",
28                 "properties": {
29                     "a": { "type": "number" },
30                     "b": { "type": "number" }
31                 },
32                 "required": ["a", "b"]
33             }
34         }
35     }
36 ]
```


Function Calling

실습

```
38 def say_hello(name):
39     return f"안녕하세요, {name}님!"
40
41 def add_numbers(a, b):
42     return f"{a} + {b} = {a + b}"
43
44 user_input = input("질문을 입력하세요: ")
45
46 response = client.chat.completions.create(
47     model="gpt-4o",
48     messages=[{"role": "user", "content": user_input}],
49     tools=tools,
50     tool_choice="auto"
51 )
52
53 tool_call = response.choices[0].message.tool_calls[0]
54 name = tool_call.function.name
55 args = json.loads(tool_call.function.arguments)
56
57 if name == "say_hello":
58     result = say_hello(**args)
59 elif name == "add_numbers":
60     result = add_numbers(**args)
61 else:
62     result = "알 수 없는 함수입니다."
63
64 print("결과:", result)
```

Function Calling

채팅 서비스에 Function Calling 기능 구현

- Day1 ~ Day5에 구현했던 @chatbot, @image, @tts 기능을 Function Calling을 사용하여 하나의 태그 @gpt로 병합하여봅시다.

Function Calling

채팅 서비스에 Function Calling 기능 구현

Chat

tester:
@gpt 지피티 안녕?
chatbot:
안녕하세요! 😊 무엇을 도와드릴까요? 궁금한 것이나 요청하실 내용이 있으면 말씀해 주세요.
tester:
@gpt 고양이 그림 그려줄래?
imagebot:



Send

파일 선택 선택된 파일 없음

Send Image

Start STT

Start Talk

Stop & Send

Chat



tester:
@gpt 안녕하세요를 음성으로 바꿔줘!
TTS:



0:00 / 0:00



음성 : “안녕하세요”

Send

파일 선택 선택된 파일 없음

Send Image

Start STT

Start Talk

Stop & Send

Function Calling

@calendar

- @calendar 기능도 일종의 Function Calling으로 볼 수 있을까요?
 - 자연어로 일정에 관한 명령을 입력
 - GPT를 통해 일정을 생성/변경/삭제 할지, 시간과 어떤 이름으로 일정을 추가할지 JSON의 형태로 받음
 - JSON에 있는대로 코드 실행.