

본선 진출팀 여러분을 위해 안내사항을 업데이트하는 문서입니다.
주요 업데이트 사항이 발생하면, 오픈채팅방으로 리마인드 보낼게요.

문의: 카카오 조훈(오픈채팅: <https://open.kakao.com/o/gisz2X2h>)

<목차>

본선 안내	1
카나나 API	1
Kanana-2-30b <update: 2025-12-04>	1
Kanana-1.5-v-3b	3
PlayMCP	5
주요 변경사항: <update: 2025-12-05>	5
MCP G/W 접근용 인증 토큰 (AccessToken. 인증 토큰을 직접 사용하는 방법)	6
MCP G/W OAuth Client (OAuth 인증 과정을 모두 구현하는 방법)	8
MCP Gateway 사용 안내	9
1. 인증 토큰을 직접 사용하는 방법 (제한된 카카오 계정. 심플)	9
2. OAuth 인증 과정을 모두 구현하는 경우 (어려움)	10
MCP Gateway OAuth2.1 지원 사양	12
연결 테스트 방법	12

<본문>

본선 안내

- 일정: 2025-12-17(수)
- 장소: 여수엑스포컨벤션센터 내 워크샵룸(룸 번호 미정)
- 기타: 상세 일정표와 장소는 확정되는데로 업데이트할게요.

카나나 API

Kanana-2-30b <update: 2025-12-04>

- url(new): <https://kanana-2-30b-a3b-s7nyu.a2s-endpoint.kr-central-2.kakaocloud.com/v1>
 - url(outdated)
<https://kanana-2-30b-a3b-s7nyu.a2s-endpoint.kr-central-2.kakaocloud.com/openai/v1>
- api_key:
KC_IS_39DCHwz3U3Og48mP4I8GAiQIH1rzEOiWAHXyk132RcPnsac4fWXJc10JLjl0suE

API 사용 예시:

```
Python
from openai import OpenAI
import json

API_KEY =
'KC_IS_39DCHwz3U30g48mP4I8GAiQIH1rzE0iWAHXyk132RcPnsac4fWXJc10JL1jl0suE'
client =
OpenAI(base_url="https://kanana-2-30b-a3b-s7nyu.a2s-endpoint.kr-central-2.kakao
cloud.com/openai/v1", api_key=f"{API_KEY}")

response = client.chat.completions.create(
    model=client.models.list().data[0].id, # kanana-1.5-32.5b-instruct
    messages=[
        {"role": "system", "content": "당신은 카카오(kakao)에서 개발된 친절한
인공지능 언어모델이고 이름은 카나나(kanana)입니다. 2024년 7월 이후 사건에 대한 정보는 알
수 없다고 답해야합니다. 현재 시간, 날짜, 사건 등 외부 정보를 참조해야 답할 수 있는 질문에는
외부 검색을 사용하라고 추천하세요. URL에 기반한 사용자 질의의 경우 사용자에게 URL에 있는
정보를 직접 입력하도록 요청합니다. 카나나(kanana)의 모델 사이즈나 파라미터 정보는
비공개입니다."},
        {"role": "user", "content": "이순신에 대해 알고 있나요?"},
        {"role": "assistant", "content": "네 알고 있습니다"},
        {"role": "user", "content": "이순신에 대해 알려줘"}
    ],
    temperature=0,
    max_tokens=256
)

print(response.choices[0].message.content)
```

툴콜 사용 예시:

```
Python
from openai import OpenAI
import json
```

```

API_KEY =
'KC_IS_39DCHwz3U3Og48mP4I8GAiQIH1rzEOiWAHXyk132RcPnsac4fWXJc10JLljl0suE'
client =
OpenAI(base_url="https://kanana-2-30b-a3b-s7nyu.a2s-endpoint.kr-central-2.kakaocloud.com/openai/v1", api_key=f"{API_KEY}")

tools = [{
    "type": "function",
    "function": {
        "name": "getCurrentTimeForLocation",
        "description": "특정 지역의 현재 시간 정보 제공",
        "parameters": {
            "type": "object",
            "properties": {
                "location": {"type": "string", "description": "시간 정보를 가져올 지역 이름"}
            },
            "required": ["location"]
        }
    }
}]

response = client.chat.completions.create(
    model=client.models.list().data[0].id,
    messages=[{"role": "user", "content": "Honolulu 지금 몇시야?"}],
    temperature=0.1,
    tools=tools,
    tool_choice="auto",
)

tool_call = response.choices[0].message.tool_calls[0].function
print(f"Function called: {tool_call.name}")
print(f"Arguments: {tool_call.arguments}")

```

Kanana-1.5-v-3b

- url: <https://kanana-v-3b.a2s-endpoint.kr-central-2.kakaocloud.com/v1/chat/completions>
- api_key:
KC_IS_39DCHwz3U3Og48mP4I8GAiQIH1rzEOiWAHXyk132RcPnsac4fWXJc10JLljl0suE

Python

```
import base64
from openai import OpenAI

openai_api_key =
f"KC_IS_39DCHwz3U30g48mP4I8GAiQIH1rzEOiWAHXYk132RcPnsac4fWXJc10JLljl0suE"
openai_api_base =
"https://kanana-v-3b.a2s-endpoint.kr-central-2.kakaocloud.com/v1"
model = "kanana-1.5-v-3b"

client = OpenAI(
    api_key=openai_api_key,
    base_url=openai_api_base,
)

# Single-image input inference
def run_single_image() -> None:
    with open('./test.jpeg', "rb") as file:
        image_base64 = base64.b64encode(file.read()).decode("utf-8")
    chat_completion_from_base64 = client.chat.completions.create(
        messages=[{
            "role": "user",
            "content": [
                {
                    "type": "text",
                    "text": "What's in this image?"
                },
                {
                    "type": "image_url",
                    "image_url": {
                        "url": f"data:image/jpeg;base64,{image_base64}"
                    }
                },
            ],
        }],
        model=model,
        max_completion_tokens=2048,
        extra_body={"add_generation_prompt": True, "stop_token_ids": [128001]},
    )

    result = chat_completion_from_base64.choices[0].message.content
    print("Chat completion output from base64 encoded image:", result)

if __name__ == "__main__":
    run_single_image()
```

PlayMCP

주요 변경사항: <update: 2025-12-05 >

1. PlayMCP의 MCP Gateway 인증을 연동하는데에 두 가지 방법이 있습니다. 이 중 쉬운 방법은 인증 토큰을 직접 사용하는 방법이며, 가급적 이것을 먼저 검토하시기를 권장드립니다. OAuth 인증 과정을 모두 구현하는 방법은 OAuth 인증에 있어 전문적인 지식을 요합니다.
2. 현재 MCP Gateway는 PlayMCP의 사용자 도구함과 연결됩니다. OAuth 인증 플로우를 모두 구현하시는 경우, `grant_type=Authorization_Code` 으로 동작하므로, 이에 맞춰 개발하시기 바랍니다.
<https://modelcontextprotocol.io/specification/2025-03-26/basic/authorization#oauth-grant-types>
3. OAuth 인증플로우를 모두 구현하시는 경우, 기 제공해드린 `Client_id`와 `Client_Secret`을 이용해, MCP Inspector 로 연결테스트를 해보실 수 있습니다.
<https://github.com/modelcontextprotocol/inspector>
MCP 인스펙터를 로컬 환경에서 구동하고, `authentication` 영역에 `client_id`, `client_secret`을 넣으면 실제로 연결테스트를 해볼 수 있습니다.

팀명	닉네임	access_token
AX+HCI	유나니	eyJraWQioIrb21pLXJzYS1rZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWliOiIzNTc5liwYXVkljoiVJnZGxUejJMaE51VENY1dSS0za2NhWndJtJlrvJlMfPocKJsdmFaYylslm5ziI6MTc2NDg5OTU2OSwic2NvcGUioIlsZGvmYXVsdCjDl.Cjpc3MIoJodHrwcwzvl3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWVnWliwiZXhwIjoxNzY3NTczNjQxLzCjYpYXQlOjE3NjQ5MTEyNTMsImp0aSI6IjRjMjNiMwVklWFIM2EiNDgzOC1hMzE0LTcyN2U2OTdmNDlmMyJ9.nHxUZSsJ-YGSw6Txd0-iWUvjKc-m9v9J3h7dMl72WBoIc6QzNqH6_Zcn0T1Wu9FF13v2MFAfVhWtww4UTJJfTsXxAcMPOiGkREKSow2uCEs-2uFbBkTgM2hKcRvXwBzOgFRDlCGKcAkmSjN08mDd-1Ynsuz3FMRrZzhbwMnohWB_w0RrGfCGD3l6SqkAwdx8VyQ5OANAv-s75pg22V2qkd5hChafdX6d0uvgr-ls_VB899ylyMjn73-bW5LA_OVFf5BdPR0j5Z710kvVlhbW96AbPznXTh3kylXRI0YKcZQgOfMbXZvv9hDbhZ-CplCnNXlGejzgfFeFY9Q
	김준섭	eyJraWQioIrb21pLXJzYS1rZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWliOiIzNTc5liwYXVkljoiVJnZGxUejJMaE51VENY1dSS0za2NhWndJtJlrvJlMfPocKJsdmFaYylslm5ziI6MTc2NDg5OTU2OSwic2NvcGUioIlsZGvmYXVsdCjDl.Cjpc3MIoJodHrwcwzvl3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWVnWliwiZXhwIjoxNzY3NTc5Mzg2LzCjYpYXQlOjE3NjQ4OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EiNDgzOC1hMzE0LTcyN2U2OTdmNDlmMyJ9.SmjeVogKytnY-Zpgga3rw3SO-BjNJ3USjQPC9-Ru0dqZXAYXQUtawBxqk-B_01HXUpv4kalyjT6CGfx4mZ3hXEPUGENZ3BW0tCylJ3wAaBmWkl-AVP_Hvzoo96YsStv5De7MboVwCe82yTbJwzFKiVhWMkOgCjmRgNfnnmKDNJSZKKYfYfZ46Frtj6SWNxy7NSYVi-FyJkGsrXbK-InbuxoKB29dMbvUUYQc301j8kF7J2WNNmlsnPOKwHrVbPp7HsEE0tW-blFM2_knA9hVlDlZfMYVl6EL53V_90dN9hvrclXCewuPxfHdz9tsTTUhaI_3Nsov2A7_dn8uUaMAMoHvG
	손효민	eyJraWQioIrb21pLXJzYS1rZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWliOiIzNTc5liwYXVkljoiVJnZGxUejJMaE51VENY1dSS0za2NhWndJtJlrvJlMfPocKJsdmFaYylslm5ziI6MTc2NDg5OTU2OSwic2NvcGUioIlsZGvmYXVsdCjDl.Cjpc3MIoJodHrwcwzvl3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWVnWliwiZXhwIjoxNzY3NTc5Mzg2LzCjYpYXQlOjE3NjQ4OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EiNDgzOC1hMzE0LTcyN2U2OTdmNDlmMyJ9.trZGejgZFiuSD2pKlwyuR17WPVD6vGJSYpfeM87h37jjkDL2HVASOBH746jineVwFFORulrvuxcqIjLBXmX11X_9NULnQdtsYiZVL57b4kVhn3pX6nq5pxDLTPm4biTGpQlrcPgeKlJy_PmUslA4daa7D01taObAkR72C7JxzitM1GXG2zaUQ8zJlKQcCsqyRprvz773-70hs7kwaCFCimPCPTZ9U3dNm7JdeHq9Oaensv1GBYufessUZupzuo300aUibq8SFCCOxJYOh3u8Trb42UXBwMOX330ExQVR-wwJ5iO2Y7NuLp1JHkwrkUnztKdCAu99PEJRPQZN1cg
	오른팔	윤덕규
그렉	최피	eyJraWQioIrb21pLXJzYS1rZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWliOiIzNTc5liwYXVkljoiVJnZGxUejJMaE51VENY1dSS0za2NhWndJtJlrvJlMfPocKJsdmFaYylslm5ziI6MTc2NDg5OTU2OSwic2NvcGUioIlsZGvmYXVsdCjDl.Cjpc3MIoJodHrwcwzvl3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWVnWliwiZXhwIjoxNzY3NTc5Mzg2LzCjYpYXQlOjE3NjQ4OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EiNDgzOC1hMzE0LTcyN2U2OTdmNDlmMyJ9.koh7Gg4u3ppg6dHsullb0ArzXI5SSOzaP5J_P4W8EeFNfthYe_4rcyjeAipvLtY3Fj5fGFRWewu2_4_Qrf5eO-8lIE2WwRhZtIt5a10mZQx6lQYdpKWCgKeJQLLxs13JkSUz0hNs_Gl5y4ZeG2EdLYuYjCvN_jcbJGz4dvUOuG-jGgQl6isQZJlZ07UHZzidvMZv3l7Og-JrksgfJf52lId9f35Z26y4dOYpYq2aK2P4u4dW00nNkyaRnsKCDlBW3Cn30aKlD8WVpU7w3n04l2FCCe_WhCZm8fa12L1knuKMOdRjh_VBYxody4NgQyPwcM1YEQSeezGNjZHalv4Q
	지상준	eyJraWQioIrb21pLXJzYS1rZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWliOiIzNTc5liwYXVkljoiVJnZGxUejJMaE51VENY1dSS0za2NhWndJtJlrvJlMfPocKJsdmFaYylslm5ziI6MTc2NDg5OTU2OSwic2NvcGUioIlsZGvmYXVsdCjDl.Cjpc3MIoJodHrwcwzvl3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWVnWliwiZXhwIjoxNzY3NTc5Mzg2LzCjYpYXQlOjE3NjQ4OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EiNDgzOC1hMzE0LTcyN2U2OTdmNDlmMyJ9.VopiTAdfriXRC-y_3Vn0Yx62U3JvnbJORZC14i2TWF3elypmOY6uzjmbvteTl6061vm-zZa7rH99zVxikOR0r1uAI2LkbW3l6kE53ehPnXEc2v_Z6caltpjIMyud_OKUG7jPFxc_KJsrBvnTB8RMKZ-OXifXUjFKfjztSpQ1BOUpKryNXNHiyENOp6Gk-xP4GuhccKcDENLZ2l6RghKLqa6TzHmrfGfJ_ZseOX1HhcvKwHXBZKXmCk_ZiUrNDl7QyM7eTa5Bm7YsdW8dWxo6ky9TNIgiZJLWwtv9432xNcQfau7R_mvVa9lhd4aQ3RpJvXy9_163BU4DXrxK6A
VeriVeri스트로Veri	최정혁	eyJraWQioIrb21pLXJzYS1rZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWliOiIzNTc5liwYXVkljoiVJnZGxUejJMaE51VENY1dSS0za2NhWndJtJlrvJlMfPocKJsdmFaYylslm5ziI6MTc2NDg5OTU2OSwic2NvcGUioIlsZGvmYXVsdCjDl.Cjpc3MIoJodHrwcwzvl3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWVnWliwiZXhwIjoxNzY3NTc5Mzg2LzCjYpYXQlOjE3NjQ4OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EiNDgzOC1hMzE0LTcyN2U2OTdmNDlmMyJ9.VopiTAdfriXRC-y_3Vn0Yx62U3JvnbJORZC14i2TWF3elypmOY6uzjmbvteTl6061vm-zZa7rH99zVxikOR0r1uAI2LkbW3l6kE53ehPnXEc2v_Z6caltpjIMyud_OKUG7jPFxc_KJsrBvnTB8RMKZ-OXifXUjFKfjztSpQ1BOUpKryNXNHiyENOp6Gk-xP4GuhccKcDENLZ2l6RghKLqa6TzHmrfGfJ_ZseOX1HhcvKwHXBZKXmCk_ZiUrNDl7QyM7eTa5Bm7YsdW8dWxo6ky9TNIgiZJLWwtv9432xNcQfau7R_mvVa9lhd4aQ3RpJvXy9_163BU4DXrxK6A
	김은채	eyJraWQioIrb21pLXJzYS1rZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWliOiIzNTc5liwYXVkljoiVJnZGxUejJMaE51VENY1dSS0za2NhWndJtJlrvJlMfPocKJsdmFaYylslm5ziI6MTc2NDg5OTU2OSwic2NvcGUioIlsZGvmYXVsdCjDl.Cjpc3MIoJodHrwcwzvl3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWVnWliwiZXhwIjoxNzY3NTc5Mzg2LzCjYpYXQlOjE3NjQ4OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EiNDgzOC1hMzE0LTcyN2U2OTdmNDlmMyJ9.xHEgg1u9cNvt0_4D2wFk_ORtyvrH0MLhJUGZQ9z-wyCx-8h-6Vs-e_W-ymvPjQl3uW6c-7IM5LY5-xJp3AqmvDhgQ4lStiGj7OL5Ye7K3VCqVb7WoynidmrSqk52ndnmdDnyp1wt9a9bxbzPncPv7CnQ9S6ff7SE02Y8oLbqGsX-VXGIBBMjriXA9Y7l8lRkVjwoYDe7DvJtWkCpXknblMTQ-8xtiP72AJGXJ8HrWUcmcTQM5b12918r3zhalBrcQ475FD7OLWYpU7w3n04l2FCCeBd91Vgv1DapxA9CMHoA5syMkw6U7lW6BIZzhwm5aydt4k35ozuQIKTgrmc-8qegga3A
	정민호	eyJraWQioIrb21pLXJzYS1rZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWliOiIzNTc5liwYXVkljoiVJnZGxUejJMaE51VENY1dSS0za2NhWndJtJlrvJlMfPocKJsdmFaYylslm5ziI6MTc2NDg5OTU2OSwic2NvcGUioIlsZGvmYXVsdCjDl.Cjpc3MIoJodHrwcwzvl3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWVnWliwiZXhwIjoxNzY3NTc5Mzg2LzCjYpYXQlOjE3NjQ4OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EiNDgzOC1hMzE0LTcyN2U2OTdmNDlmMyJ9.OpNpEr0ypcJhqS-bxnrfuYqQRQEx-m4zv2f7uGaeBPYdPXTPIKGa1G7Vwve0vZF5hwifRYbc00evExgkEMH0cB7vWxGfTNAkpYyrO9P-focqfqr1hlib0dFsmba6ZegL_b2ta_AZPZb61lgQ0arDl1DORQ94bh_UbIil33MMWwPhqqzC0uHznDsBhhcRwktTfJE_z44Kqwi5MD1Tx6Qwi50DLrqvQZCT8RZBZUroQ6KH0CeyfJzEtZluWQmBkdKdJqRQ4bCbtm3ip5q8m8w7gn1MBXCO0uYjHbJ4clMetU3vvXKSaK7GoKbL5sykfjtc0FLFYa7L-SkiePLH5kEuJ2thg

	남서용	eyJraWQiOiJrb21pLXJzYS1rZXkiLCJhbGciOiJSUz11NiJ9.eyJzdWliOiZmZyY0liwiYXVkljoiQkUemNPSzdJQURCmxdHhrd0N1MTNFa0ZVcDJRWWEYzODZkXzQtc0VOCiSlm5iZiI6MTc2NDg5OTU2OSwic2NvcGUiOlsiZGVmYXVsdCJdLClCjpc3MiOiJodHRwczovL3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWNwliwiZXhwIjoxNzY3NTc5Mzg2LClpYXQiOjE3NjQ0OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EtNDgzOC1hMzE0LTcyN2U2OTdmNDImMyJ9.3t1fNa5QYXons-BTG-PJ06NZyYJWqE_lItHOG_Dhtw8qr0OBzAOBeoDmjXEQYoiQfqVhv8i aL-dEIQKGdsYiBWr8nuLA5KgytZy25paeVF9fqDI1SSLeXdKe74fMhKAYdmcGazw7DmVQc924fMMcCCSxurh7G_algobb8PxAt49-l4H ia9SClqE7-JPFGKbhRcyz3YntR_og7ZhqWiW1NU0XF4gaTdeUBDj9g0Z2iGWmdGbZHFMA3ihK4_00DZIMWKF02Tu7zMmElBc0FO 3PeOEeYd2e7RY11fivkbGosoqsmZNo8JnHRH2TjERGKEfOOXxy_EKzbu98coSjyHA
안심톡톡 (양방향보안)	온다	eyJraWQiOiJrb21pLXJzYS1rZXkiLCJhbGciOiJSUz11NiJ9.eyJzdWliOiI0MDA4liwiYXVkljoiUgXR0Q0UGVvNXIGSU5lakp3U0pqN0NkcViHdjB5aWxHX0FsSU1xNGtoNCiSlm5iZiI6MTc2NDg5OTU2OSwic2NvcGUiOlsiZGVmYXVsdCJdLClCjpc3MiOiJodHRwczovL3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWNwliwiZXhwIjoxNzY3NTc5Mzg2LClpYXQiOjE3NjQ0OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EtNDgzOC1hMzE0LTcyN2U2OTdmNDImMyJ9.zWC5WD0vNJVhHpdT86c4tIRIBTgQyYIMC6yDIXh6iVqB6oCbz3_YyQ8AAw-qHqwr9FWbK21VWQ-74WK1uZOqgpcvk99LrPKDcg57I7OHg8LPQR_Zvt_oZhDvpv0hJJGodh93epK2unPZxh8v_R6mQ4vS80gCBrk05K6O0Ev884aS6PnRFjogdC2NQ2dqChqyDJNGS1Ap1CcH6dKPe5FxZ3caB-muXP9xlKOzN6Uuly2yKQdxa7ThRlg1acjIBgqeIFEsxDuqZnbnm30cviFAP4UU_LE9PHIJhB_Ts1BCx40tz1N5NCw0_WoT4D4s7wDIBCAoAus03koxdl2KGHRQ
FireArm	윤형준	eyJraWQiOiJrb21pLXJzYS1rZXkiLCJhbGciOiJSUz11NiJ9.eyJzdWliOiZmZyY0liwiYXVkljoiMXVZVmFVWWRR5My16N3NMQMf5UDNDUjhRb3hZLU9wVVRSTEJFMzMakJ3ZyIsIm5iZiI6MTc2NDg5OTU2OSwic2NvcGUiOlsiZGVmYXVsdCJdLClCjpc3MiOiJodHRwczovL3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWNwliwiZXhwIjoxNzY3NTc5Mzg2LClpYXQiOjE3NjQ0OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EtNDgzOC1hMzE0LTcyN2U2OTdmNDImMyJ9.mDmRcwQTqdbgT_j_1VDaln0arGL-KqZQya4QK8s70MDVdsZ1RUt7KGqpxkaUGISzUIJfesZDU3Tw0r_z3-colDsN60R3wSNdm015NmTKKeuO_h4VA7SuViV8GsEVgj0xYnQRy8IXH5sWCzo1KtDv6n7Poy3KoHvxA8Jk2ZXMQNcYY7Htsi5K1W6Pkzjz7KXZcD25_zjJmIEo03lg016JXdxgECNGM-7XW1ciiCS6fWs5Ri1YL36g_vkiOc_HX6T1SC7BchlA3r_TaEulrv3wQGiZkgIT21d_lmx98AzbXr_oflAcTyAs01LgQGZFeN97_ZmsXC1Piifj5Rzk-MgQ
	전도윤	eyJraWQiOiJrb21pLXJzYS1rZXkiLCJhbGciOiJSUz11NiJ9.eyJzdWliOiZmZyY0liwiYXVkljoiMXVZVmFVWWRR5My16N3NMQMf5UDNDUjhRb3hZLU9wVVRSTEJFMzMakJ3ZyIsIm5iZiI6MTc2NDg5OTU2OSwic2NvcGUiOlsiZGVmYXVsdCJdLClCjpc3MiOiJodHRwczovL3BsYXlhdXRoLmtha2FvLmNvbS9wbGF5bWNwliwiZXhwIjoxNzY3NTc5Mzg2LClpYXQiOjE3NjQ0OTk1NjksImp0aSI6IjRjMjNiMwVklWFIM2EtNDgzOC1hMzE0LTcyN2U2OTdmNDImMyJ9.Jmje22R-6B7v4jh6uoXNJMSelu4RgcrHiOnAjkn40XAW9EkLLOft-eWzBgGP3j5WzNIP1GMuJunzQUKAJi8_eeXaBP5FIC97puTQHdVH0ZYAn0Ojeq7Z0KXkc2dHNKw36-V564NMKVuUpS3B21FGHmGJiqkG27TAdaufmO1fz4_UsbU_70LpELOKcHcjsXFN4z8SqC6XZTxASXhXwPkt8mWcFe965q9UJa3UfoMGn64hdH8bm9QMLikWrPsVgmm9m2muvPe3Uf0Z9XMWA83bzA1cgijjMdwJvaiKT9tA9KvayGeBzdM_m14eChxNBkCtyQ-tSFHct-UkNP1Y2g
논대워	안연수	

MCP G/W OAuth Client (OAuth 인증 과정을 모두 구현하는 방법)

팀명	client_id	client_secret
AX+HCI	Lh_D6V2QhTaIWRI0HgyAKyoCQ TGBsUF3EipCbA8JFwQ	s2wh2f1b6XoNTUYnWeY_17xASiBiAeMRaYAI dbESXIkctL080hFszYKndd7nWidG
오른팔	V2gdITz2LhNuTCXcWei-3kcaZwl NYkv9I0ZNRBRvaZc	B8PAA4CEj7Uh9edOsoxCW8u237aWllo2r4_PU F1dXhfhI2BLy27loIRkfJ29cQq7e
그렉	hvGX9dohuw-mkLhStZoVSehF7i Nnz-gqsRGkQ9MsvvY	XMMtSsrXiD5prN-INyFDI39pg6LXD3bG0BaiSZ hMTWfOsN0zRBZp3hxA0SgpxP8
VeriVeri스트 로Veri	5BI-qjchGRxHKr2D7I5ZMRvFyfQ Ddm6T5M2zOHM6bdo	L5GmaGnXltWUdetxgrlRTQG0bdk8nek8681Jig JPYrJB6-MMZi5fjufpij84grvY
카카오비전	BITzcOK7IADBrlktxkwCu13EkFU p2QXF386d_4-sE_8	J95lg7e-bezuws6XQUuCxOrBYKrvdoQCiwJ9uJ DiE0F9IishcNYz4GtKfi9UpkTX
안심톡톡 (양방향보안)	mH1GD4Peo5yFINejJwSJj7CdqY Gv0yilG_AlIMq4kh4	UtxTkBRVykJ2GbPGQ14N_041j_buBqjXlc8oqt ekOJh-MUJIDCSvzQUkbEKiTssG
Firearm	1uYVaUYdy3-z7sLBayP3CR8Qox Y-OpUTRLBE36LjBwg	29ELwarMzJT-hrrmUGPYOXVjOMSI1qJZjUMm jcxgEpRWn7nhKVuXp04t7T5myenW
논대워	SnDLNkgXMLMsGatQfTVeAD9N 8yajvMIUDmGICtHMvCA	N9AiLf77EHUQrFCSYcTEd3LPV0azHCLn3XeS o2xGwcgHGadEiRgauUbsiM8K5b7S

MCP Gateway 사용 안내

본 문서는 정보과학회 학술대회 참여자 분들을 대상으로 제한적으로 제공되는 내용이며, 무단으로 유출되는 것을 금합니다.

MCP Gateway 는 PlayMCP에 등록된 여러 MCP 서버를 모아 단일한 MCP 서버로서 도구 리스트를 제공합니다. 학술대회에 참여하시는 팀에서는 MCP 서버를 이용해 AI 에이전트 개발 등에 활용하실 것으로 예상하고 있습니다. MCP Gateway와 연동하기 위한 방법과 최소한의 가이드를 제공해드립니다.

1. 인증 토큰을 직접 사용하는 방법 (제한된 카카오 계정. 심플)

가장 간단한 방법으로, 에이전트 개발 등의 과정에서 카카오 MCP Gateway 와 통신하기 위한 액세스 토큰을 카카오에서 직접 제공해드리는 방식입니다. 이를 위해 참가자 분들은 (1) PlayMCP에 접속 및 가입하고 (2) 가입하신 카카오 계정의 이메일 주소와 PlayMCP에서의 닉네임을 제공해주셔야 합니다.

이후, 카카오에서는 전달받은 카카오 계정을 기준으로 MCP Gateway 접근을 위한 액세스 토큰을 발행하고, 이를 참가자 분들에게 전달드립니다. 이 토큰은 학술대회 참가자를 위해 임의 발행된 토큰으로 30일 간의 유효 기간을 갖습니다.

아래는 액세스 토큰을 이용하여 MCP 서버에 접근하는 샘플입니다.

[Initialize 요청 단계](#) 예시

```
curl --location 'https://playmcp.kakao.com/mcp' \
--header 'Authorization: Bearer <전달받은 액세스 토큰>' \
--header 'accept: application/json, text/event-stream' \
--header 'Content-Type: application/json' \
--data '{
  "method": "initialize",
  "params": {
    "protocolVersion": "2025-06-18",
    "capabilities": {
      "sampling": {},
      "elicitation": {},
      "roots": {
        "listChanged": true
      }
    },
    "clientInfo": {
      "name": "mcp-inspector",
      "version": "0.16.5"
    }
  },
}
```

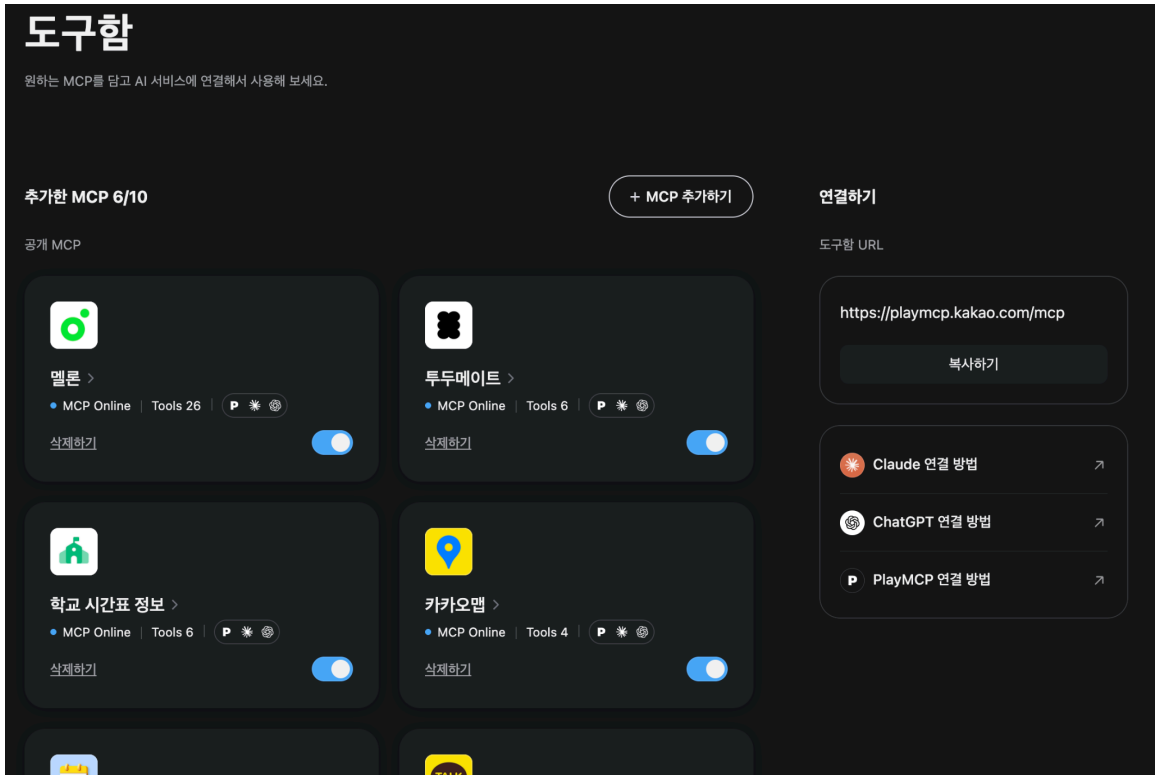
```
"jsonrpc": "2.0",
"id": 0
}
```

```
gtpark@gtparkui-MacBookPro ~ % curl --location 'https://[redacted].playmcp.kakao.com/mcp' \
--header 'Authorization: Bearer eyJraWQiOiJrb21pLXJzYS1rZXkiLCJhbGciOiJIUzI1NiJ9.eyJzdWI0IjoiIiwiaWF0IjoiUDZStklyW
[redacted]
RoWcS9EeRbZR08HrcPDzQbXUb0d403d_7nQ3qpnQEeXwnRw' \
--header 'accept: application/json, text/event-stream' \
--header 'Content-Type: application/json' \
--data '{
    "method": "initialize",
    "params": {
        "protocolVersion": "2025-06-18",
        "capabilities": {
            "sampling": {},
            "elicitation": {},
            "roots": {
                "listChanged": true
            }
        },
        "<....
{"id":0,"result":{"capabilities":{"tools":{}},"protocolVersion":"2025-06-18","serverInfo":{"name":"mcp-gateway","v
ersion":"1.0.0"},"title":"PlayMCP Gateway"},"instructions":"이 MCP 서버는 카카오페이 공식적으로 제공하는 안전한 MCP
서버입니다. 본 MCP 서버는 사용자의 정보 보호를 최우선으로 생각합니다."},"jsonrpc":"2.0"}%`
```

<그림. cURL 커맨드를 이용하여, MCP Gateway Initialize 수행하고 결과를 받는 예시>

MCP Gateway 서버와 간단히 연동하여 개발하실 분들은 위와 같이 카카오 액세스 토큰을 직접 하드코딩하여, 에이전트 등의 개발에 활용하시기 바랍니다. MCP 서버의 도구 호출을 위한 세부 내용은 [MCP 명세 문서](#)를 참고하시기 바랍니다.

MCP Gateway는 [PlayMCP의 도구함](#)에 담긴 MCP 서버들이 연동되어 제공됩니다. 따라서, 전달주신 카카오 계정에 에이전트 개발에 필요한 도구들을 담고, MCP Gateway와 연동해서 사용해야 합니다.

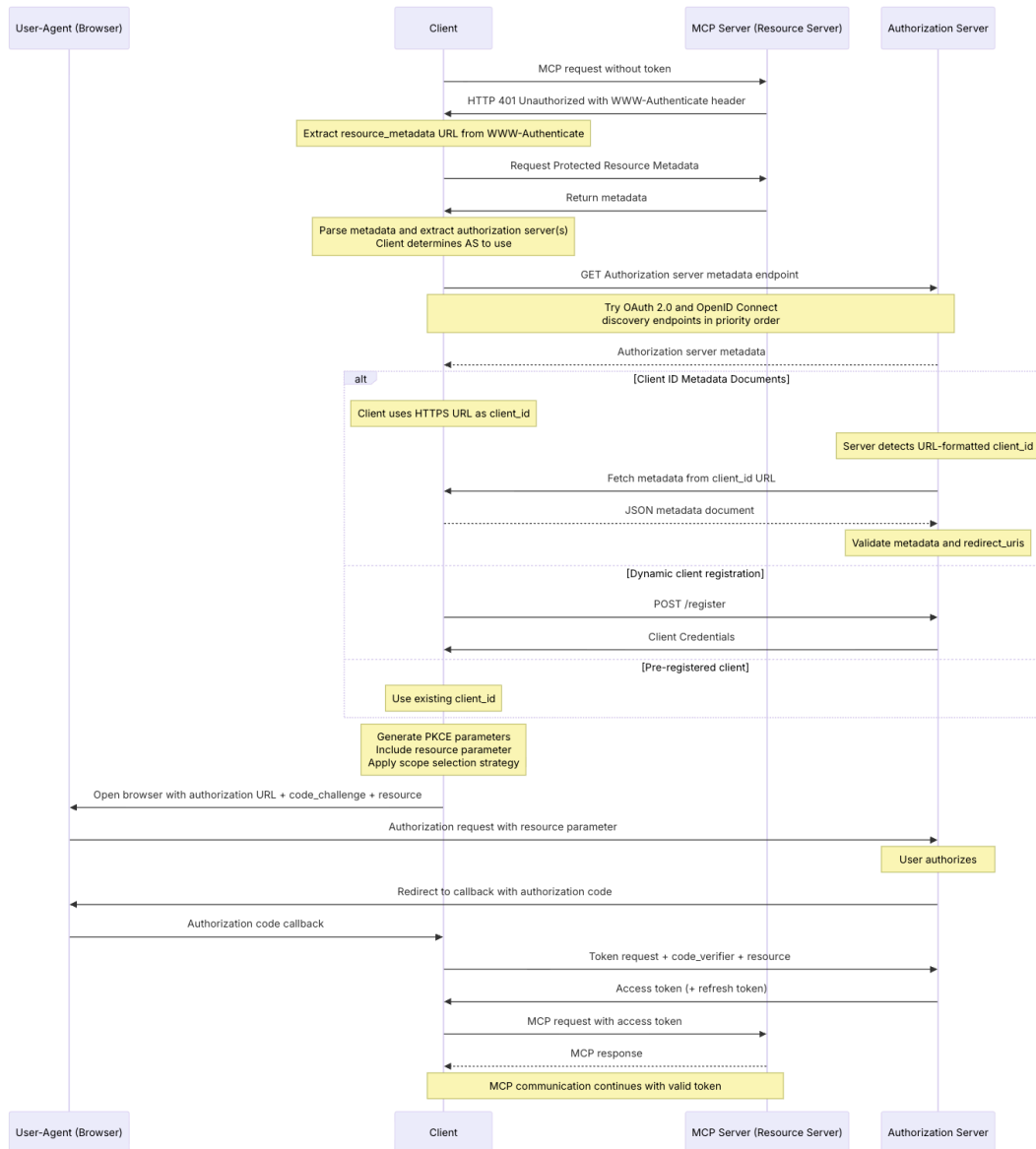


<그림. PlayMCP 도구함>

2. OAuth 인증 과정을 모두 구현하는 경우 (어려움)

보다 어려운 방법으로, MCP Gateway에 OAuth 인증을 정식으로 연동하여 테스트 하는 방식입니다. 프로젝트 내에서 OAuth 인증을 연동하고자 하실 경우, (1) 카카오에서는 **OAuth ClientId, ClientSecret** 등을 생성하여 전달드리게 됩니다. (2) 이후 **OAuth** 인증 과정에서의 클라이언트 엔드포인트로서, **redirect_uri**를 작성하여 카카오에 전달해주셔야 합니다.

OAuth 인증 방식을 채택하는 경우 [OAuth 2.1 인증 플로우를 모두 구현](#)해야 합니다. 이 플로우는 OAuth 표준 스펙을 따르며, 추상적으로 (a) 초기 MCP 서버 요청 및 401 응답 수신 (b) Metadata 를 통해 인증서버 정보 획득 (c) 토큰 교환의 시작 (d) 인증 코드 발행 및 **redirect_uri**로의 페이지 이동 (e) 액세스 토큰 발행 과 같은 과정을 거쳐야 합니다.



이 과정은 OAuth 인증에 대한 깊은 이해를 필요로 하므로, OAuth 인증에 대한 전문적인 지식이 없는 경우 [1. 제한된 카카오 계정을 사용하는방법](#)을 이용하시기 바랍니다.

MCP Gateway OAuth2.1 지원 사양

- grant_type=authorization_code
- scope=default

연결 테스트 방법

아래는 전달받은 OAuth Client 정보로 OAuth 인증 과정에 대한 연동 과정을 모두 지원하는 MCP Inspector로 테스트해보는 과정입니다. 이를 구현하려면 인증 플로우를 모두 구현해야 하므로, [1. 인증 토큰을 직접 사용하는 방법](#)을 활용하시는 것을 추천드립니다.

1. [MCP Inspector 설치](#) (로컬 환경에 설치합니다)
 - a. nodejs가 설치된 상태로 아래와 유사하게 명령어 수행
 - b. `npx @modelcontextprotocol/inspector@latest`
2. MCP Gateway URL 및 기 전달받은 클라이언트 정보를 기입
 - a. url을 기입합니다.
 - b. client_id, client_secret을 입력합니다.
 - c. redirect_url은 현재 <http://localhost:6274/oauth/> 로 설정되어있으며, 이 방식을 이용할 경우 자체 redirect_url을 구성하여 카카오에 전달해주셔야 합니다.
 - d. 마지막으로 Open Auth Settings 클릭합니다.

MCP Inspector v0.17.5

Transport Type
Streamable HTTP

URL
<https://playmcp.kakao.com/mcp>

Connection Type
Via Proxy

Server Entry Servers File

Authentication

Custom Headers JSON + Add

No custom headers configured
Click "Add" to get started

OAuth 2.0 Flow

Client ID
Lh_D6V2QhTatWRz0HgyAKyoCQTCB

Client Secret
s2wh2F4b6XoNTUxwdeV_13

Redirect URL
<http://localhost:6274/oauth/>

Scope
Scope (space-separated)

Configuration

Connect

Connect to an MCP server to start inspecting

Need to configure authentication? Open Auth Settings

3. Quick OAuth Flow 클릭. 카카오 로그인에 되어있지 않다면 로그인 창이 뜰 수 있습니다.

MCP Inspector v0.17.5

Transport Type

Streamable HTTP

URL

https://playmcp.kakao.com/mcp

Connection Type

Via Proxy

Server Entry

Servers File

Authentication

Custom Headers

JSON

Add

No custom headers configured

Click "Add" to get started

OAuth 2.0 Flow

Client ID

Authentication Settings

Back to Connect

Configure authentication settings for your MCP server connection.

OAuth Authentication

Use OAuth to securely authenticate with the MCP server.

Guided OAuth Flow

Quick OAuth Flow

Clear OAuth State

Choose "Guided" for step-by-step instructions or "Quick" for the standard automatic flow.

OAuth Flow Progress

Follow these steps to complete OAuth authentication with the server.

Metadata Discovery

OAuth Metadata Sources

Client Registration

Registered Client Information

playauth.kakao.com/playmcp/oauth2/consent?scope=default&client_id=Lh_D6V2QhTalWRI0HgyAKyo

ax_hci에
PlayMCP 연결하기

ax_hci에서 PlayMCP를 통해 더 많은 기능을 사용할 수 있도록 다음 권한을 요청합니다.

✓ PlayMCP 계정 정보 확인

사용자의 PlayMCP 계정을 확인할 수 있습니다.

✓ 도구함의 MCP 및 도구 목록 확인

PlayMCP 도구함에 담긴 MCP 정보 및 도구 목록을 확인할 수 있습니다.

✓ 도구 호출 및 결과 확인/저장

PlayMCP 도구함에 담긴 MCP의 도구 호출 결과를 확인하고 저장할 수 있습니다.

✓ [필수] 개인정보 제3자 제공 동의

보기

✓ [필수] 개인정보 국외 이전 동의

보기

취소하기

모두 동의 후 계속하기

4. 이후 정상적으로 OAuth 인증이 완료되면, Connect 버튼을 누르고 Tools의 list 등 조회가 가능해집니다. 각 사용자별로 조회되는 tools/list 는 [playmcp 도구함](#)에 담긴 MCP 서버의 도구가 출력됩니다.

The screenshot displays the MCP Inspector v0.17.5 interface. On the left, the configuration panel shows the Transport Type set to 'Streamable HTTP', the URL as 'https://playmcp.kakao.com/mcp', and the Connection Type as 'Via Proxy'. The OAuth 2.0 Flow section includes fields for Client ID, Client Secret, Redirect URL, and Scope. The bottom of the left panel shows a 'Connected' status for the 'mcp-gateway' server.

The main panel is divided into two sections. The top section, titled 'Tools', contains a search bar and a list of available tools. The tools listed are:

- ✓ KakaotalkCal-GetCurrentTime: Fetches the current date, time, and day of the week. Use this when the current time is needed.
- ✓ KakaotalkCal-GetEvent: 달력에서 사용자의 일정을 조회합니다. 조회범위를 지정하지 않으면 오늘 일정을 가져옵니다.
- ✓ KakaotalkCal-CreateEvent: 일정을 캘린더에 등록합니다. 제목과 시작, 종료 시각을 받아 일정을 등록합니다. 요약 정보(title, startAt, endAt)를 사용자에게 먼저 보여주고, 명확한 확인을 받은 경우에만 실행되어야 합니다. 일부 정보만 보았던 응답이나 이전 확인 내용을 근거로 자...
- KakaotalkCal-GetTask: 달력에서 사용자의 할 일을 조회합니다. 조회범위를 지정하지 않으면 오늘 할 일을 가져옵니다.
- KakaotalkCal-CreateTask: 새 할 일을 생성합니다. 사용자의 발화에 할 일과 관련된 문맥이 포함되어 있을 때 호출됩니다. 한 문장에 여러 개의 할 일이 존재한다면 톨을 여러 번 호출하여 할 일을 등록합니다. 할 일을 생성하기 전 사용자에게 내용, 마감일, 반복 여부 등을 아래 형식으...

The bottom section, titled 'History', shows a list of recent actions:

- 2. tools/list
- 1. initialize

On the right side of the interface, there are sections for 'Select a...' and 'Server N...', both of which are currently empty.