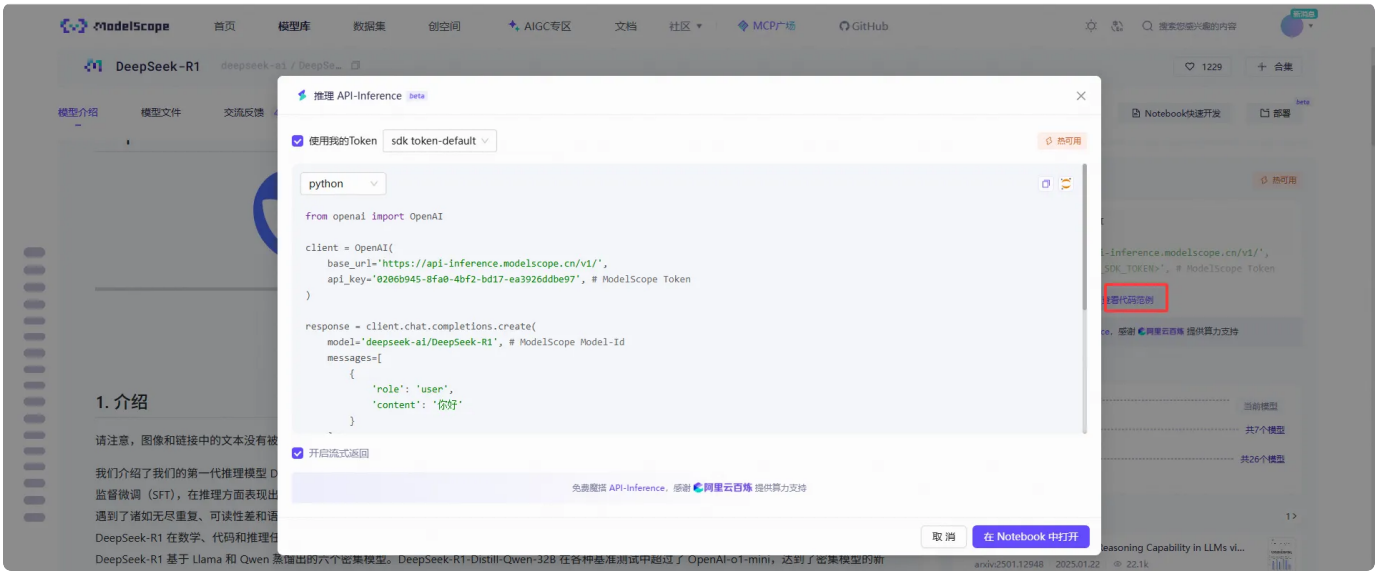
02-调用第三方平台的模型

1、魔塔社区的模型

官网地址: https://community.modelscope.cn/



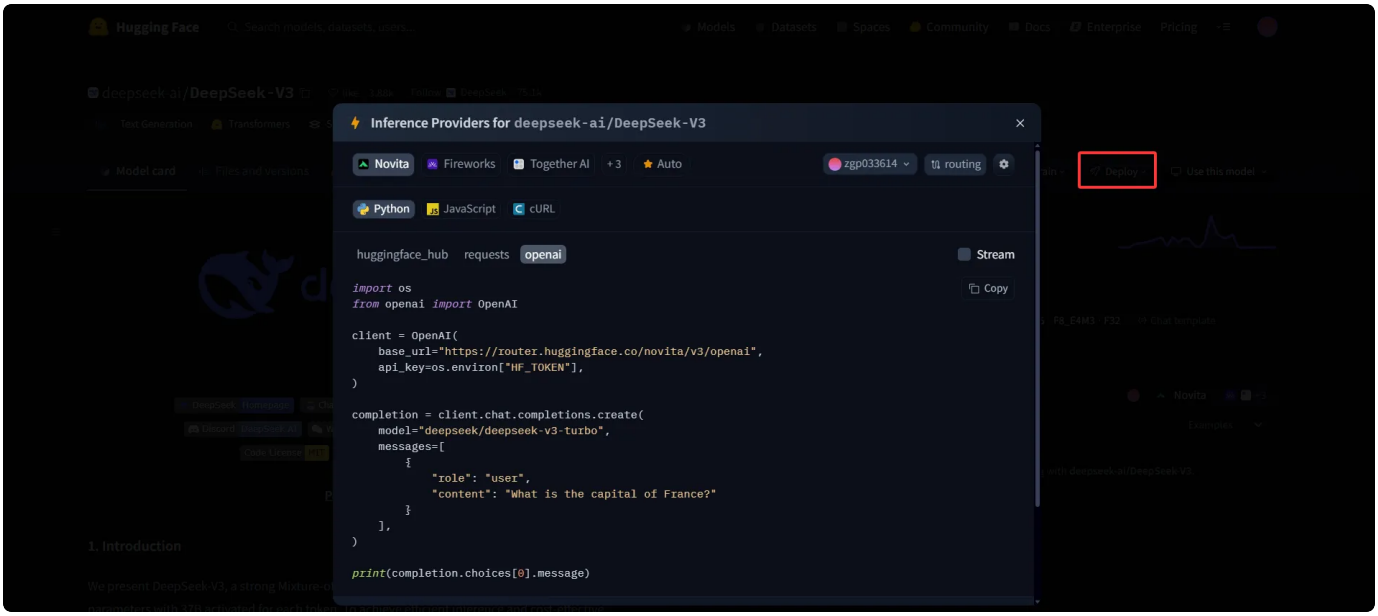
```python
from openai import OpenAI

client = OpenAI(
    base_url='https://api-inference.modelscope.cn/v1/',
    api_key='0206b945-8fa0-4bf2-bd17-ea3926ddbe97', # ModelScope Token
)

response = client.chat.completions.create(
    model='deepseek-ai/DeepSeek-R1', # ModelScope Model-Id
    messages=[
        {
            'role': 'user',
            'content': '你好'
        }
    ],
    stream=True
)
done_reasoning = False
for chunk in response:
    reasoning_chunk = chunk.choices[0].delta.reasoning_content
    answer_chunk = chunk.choices[0].delta.content
    if reasoning_chunk != '':
        print(reasoning_chunk, end='',flush=True)
    elif answer_chunk != '':
        if not done_reasoning:
            print('\n\n === Final Answer ===\n')
            done_reasoning = True
        print(answer_chunk, end='',flush=True)
```

2、HuggingFace 服务器上的大模型

官网：https://huggingface.co/



Python

```python
import os
from openai import OpenAI

client = OpenAI(
    base_url ="https://router.huggingface.co/novita/v3/openai",
    api_key  = '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',
)

stream = client.chat.completions.create(
    model="deepseek/deepseek-v3-turbo",
    messages=[
        {
            "role": "user",
            "content": "你好，能给我讲个笑话吗？"
        }
    ],
    stream=True,
)

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

3、官方服务器的模型

官网（阿里百炼平台）：https://www.deepseek.com/

```python
import os
from openai import OpenAI

client = OpenAI(
    # 若没有配置环境变量，请用阿里云百炼API Key将下行替换为：api_key="sk-xxx",
    api_key=os.getenv("DASHSCOPE_API_KEY"),
    base_url="https://dashscope.aliyuncs.com/compatible-mode/v1"
)

completion = client.chat.completions.create(
    model="qwen-plus",  # 此处以qwen-plus为例，您可按需更换模型名称。模型列表：https://help.aliyun.com/zh/model-studio/getting-started/models
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "你是谁？"}
    ],
    stream=True,
    # Qwen3模型通过enable_thinking参数控制思考过程（开源版默认True，商业版默认False）
    # 使用Qwen3开源版模型时，请将下行取消注释，否则会报错
    # extra_body={"enable_thinking": False},
)

full_content = ""
print("流式输出内容为：")
for chunk in completion:
    # 如果stream_options.include_usage为True，则最后一个chunk的choices字段为空列表，需要跳过（可以通过chunk.usage获取 Token 使用量）
    if chunk.choices:
        full_content += chunk.choices[0].delta.content
        print(chunk.choices[0].delta.content)
print(f"完整内容为：{full_content}")
```

deepseek 官网: https://www.deepseek.com/

```python
# Please install OpenAI SDK first: `pip3 install openai`

from openai import OpenAI

client = OpenAI(
    api_key='<DeepSeek API Key>'
    base_url="https://api.deepseek.com"
)

response = client.chat.completions.create(
    model="deepseek-chat",
    messages=[
        {"role": "system", "content": "You are a helpful assistant"},
        {"role": "user", "content": "你好,给我讲个笑话吧"},
    ],
    stream=False
)

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

流式输出，聊天交互模型

```python
import sys
from openai import OpenAI

client = OpenAI(api_key="sk-899fd42834ad447882e596f1787fadd7", base_url="https://api.deepseek.com")

def stream_response(messages):
    response = client.chat.completions.create(
        model="deepseek-reasoner",
        messages=messages,
        stream=True
    )

    full_content = ""
    for chunk in response:
        if hasattr(chunk.choices[0].delta, 'content') and chunk.choices[0].delta.content:
            content = chunk.choices[0].delta.content
            sys.stdout.write(content)  # 逐字输出
            sys.stdout.flush()
            full_content += content

    print()  # 换行
    return full_content

# 初始化对话
messages = [{"role": "system", "content": "你是一个幽默的助手，喜欢讲笑话和有趣的对话。"}]

while True:
    try:
        # 用户输入
        user_input = input("\n用户: ")
        if user_input.lower() in ['退出', 'exit', 'quit']:
            print("对话结束，再见！")
            break

        messages.append({"role": "user", "content": user_input})

        # 获取AI回复并流式输出
        print("AI: ", end="")
        assistant_content = stream_response(messages)
        messages.append({"role": "assistant", "content": assistant_content})

```

```
        except KeyboardInterrupt:
            print("\n对话结束，再见！")
            break
        except Exception as e:
            print(f"\n发生错误：{e}")
            break
```