

AI 駕動的 Git 提交信息

這個 Python 腳本應該放置在系統 PATH 包含的目錄中，例如 ~/bin。

```
import subprocess
import os
from openai import OpenAI
from dotenv import load_dotenv
import argparse

load_dotenv()

def gitmessageai(push=True, only_message=False):
    # 暫存所有變更
    subprocess.run(["git", "add", "-A"], check=True)

    # 獲取暫存變更的差異
    diff_process = subprocess.run(["git", "diff", "--staged"], capture_output=True, text=True, check=True)
    diff = diff_process.stdout

    if not diff:
        print(" 沒有變更可提交。")
        return

    # 準備 AI 的提示
    prompt = f"""

為以下代碼變更生成一個簡潔的提交信息，使用 Conventional Commits 格式。
使用以下類型之一：feat、fix、docs、style、refactor、test、chore、perf、ci、build 或 revert。
如果適用，請在括號中包含範圍以描述受影響的代碼庫部分。
提交信息不應超過 70 個字符。


代碼變更：
{diff}"""


    # 提交信息
    prompt += f"""
提交信息：

{diff}
    """
```

代碼變更：

{diff}

提交信息：

.....

```

# 將提示發送到 DeepSeek API
api_key = os.environ.get("DEEPSEEK_API_KEY")
if not api_key:
    print(" 錯誤：未設置 DEEPSEEK_API_KEY 環境變量。")
    return

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

try:
    response = client.chat.completions.create(
        model="deepseek-chat",
        messages=[
            {"role": "user", "content": prompt}
        ],
        max_tokens=100
    )
    if response and response.choices:
        commit_message = response.choices[0].message.content.strip()
        commit_message = commit_message.replace('`', '')
    else:
        print(" 錯誤：API 沒有回應。")
        return
except Exception as e:
    print(f"API 調用期間出錯：{e}")
    return

# 檢查提交信息是否為空
if not commit_message:
    print(" 錯誤：生成的提交信息為空。中止提交。")
    return

if only_message:
    print(f" 建議的提交信息：{commit_message}")
    return

```

```

# 使用生成的信息提交
subprocess.run(["git", "commit", "-m", commit_message], check=True)

# 推送變更
if push:
    subprocess.run(["git", "push"], check=True)
else:
    print(" 變更已提交到本地，但未推送。")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description=" 使用 AI 生成提交信息並提交變更。")
    parser.add_argument('--no-push', dest='push', action='store_false', help='在本地提交變更而不推送。')
    parser.add_argument('--only-message', dest='only_message', action='store_true', help='僅打印 AI 生成的提交信息。')
    args = parser.parse_args()
    gitmessageai(push=args.push, only_message=args.only_message)

```

然後，在你的 ~/.zprofile 文件中，添加以下內容：

```

alias gpa='python ~/bin/gitmessageai.py'
alias gca='python ~/bin/gitmessageai.py --no-push'
alias gm='python ~/bin/gitmessageai.py --only-message'

```