

AI 驱动的 Git 提交信息

这个 Python 脚本应放置在系统 PATH 包含的目录中，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):
    # 暂存所有更改
    subprocess.run(["git", "add", "-A"], check=True)

    # 获取暂存更改的差异
    diff_process = subprocess.run(["git", "diff", "--staged"], capture_output=True, text=True, check=True)
    diff = diff_process.stdout

    if not diff:
        print(" 没有更改需要提交。")
        return

    # 为 AI 准备提示
    prompt = f"""

为以下代码更改生成简洁的提交信息，采用 Conventional Commits 格式。
使用以下类型之一：feat、fix、docs、style、refactor、test、chore、perf、ci、build 或 revert。
如果适用，请在括号中包含范围以描述受影响的代码库部分。
提交信息不应超过 70 个字符。
```

代码更改：

{diff}

提交信息：

.....

```
# 将提示发送到 DeepSeek API
api_key = os.environ.get("DEEPSEEK_API_KEY")
if not api_key:
    print(" 错误：未设置 DEEPSEEK_API_KEY 环境变量。")
    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(" 错误：API 无响应。")
        return
except Exception as e:
    print(f"API 调用期间出错: {e}")
    return

# 检查提交信息是否为空
if not commit_message:
    print(" 错误：生成了空的提交信息。中止提交。")
    return
```

```

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

# 使用生成的提交信息进行提交
subprocess.run(["git", "commit", "-m", commit_message], check=True)

# 推送更改
if push:
    subprocess.run(["git", "push"], check=True)
else:
    print(" 更改已本地提交, 但未推送。")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description=" 使用 AI 生成提交信息并提交更改。")
    parser.add_argument('--no-push', dest='push', action='store_false', help='本地提交更改但不推送。')
    parser.add_argument('--only-message', dest='only_message', action='store_true', help='仅打印 AI 生成的提交信息')
    args = parser.parse_args()
    gitmessageai(push=args.push, only_message=args.only_message)

```

然后, 在您的 `~/.zprofile` 文件中添加以下内容:

```

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

```