



《OpenAI 代码自动评审组件》- 第1节-学习记录

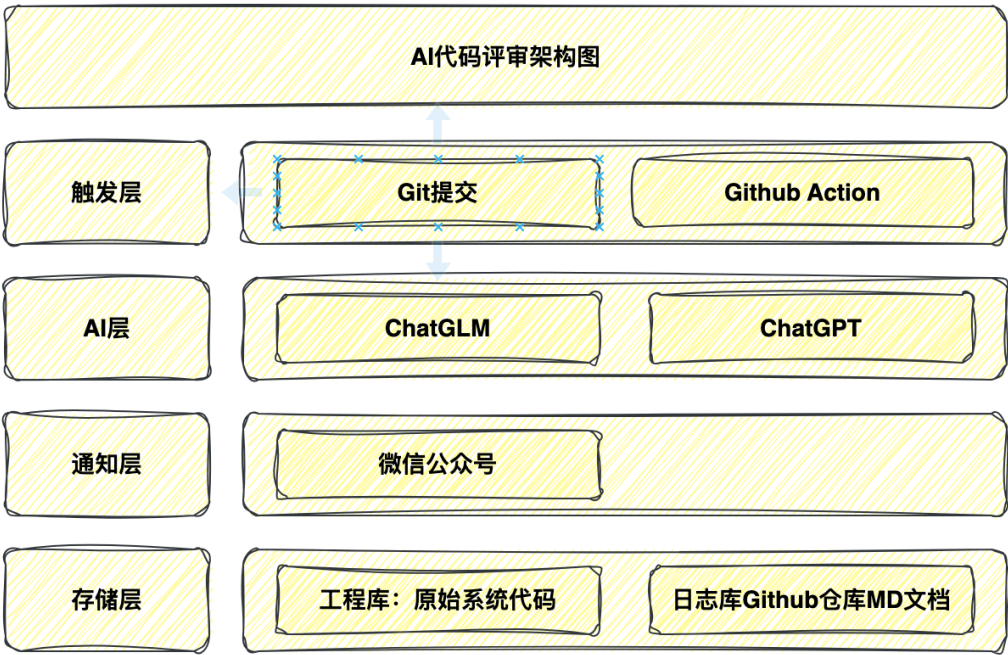
来自：码农会锁

 爱海贼的无处不在

2024年07月14日 17:23

一、本文学习记录

这个是作者的一个新的AI小组件项目，通过标题可以知道这个是通过AI进行代码自动评审的这样的一个功能。
这个组件也是作者基于 GitHub Actions + OpenAI(ChatGLM) + Git/GitHub + 公众号模板消息实现，然后串联出从代码提交获取通知，Git 检出分支变化，在使用 OpenAI 进行代码和写入日志，再发送消息通知完成整个链路。
由此，我们可以猜测到这个项目的架构图如下所示：



可以看到通过这样一个操作，完成了一个AI代码评审的功能。
作者之前已经分享过这个组件的一个基本的全貌说明，链接如下：
<https://www.51cto.com/article/792239.html>
在之前这篇文章中，作者分享了Github的脚本，这个我个人感觉是非常重要的，解决了很多人没有服务器的问题，可以通过Github Actions可以实现一些基于代码仓库的任务操作，好多开源项目也会在提交代码后，自动进行代码的单元测试自动执行，而作者解决的是代码的CodeReview问题，而这里带给我们的思考就是，当我们学会了这个后，我们可以自己进行代码提交后的任何CI/CD工作，包括单元测试、代码质量等等。
作者在第1节的视频中，给小伙伴进行了细致的需求讲解与业务分析，作者讲解了三种设计方案，以及当前采用的方案：

方案一：SDK模式

方案二：webhook模式

方案三：Github Actions方式

基于第三种设计，作者讲解了这个设计的基础流程，即开发一个Java小程序，然后在Github Actions中调用这个组件，然后将结果发送给AI进行代码评审，将AI返回的结果通知给用户，同时写入结果到Github的一个日志仓库中，保留每次的评审结果。

对于这个组件，作者的分享给大家学习到的知识如下：

- 一整套的设计方案分析和相应的技术问题处理手段，这个思考方式很重要。
- GitHub Actions 的使用机制，它的一些超级强大的用途和使用方式。
- OpenAI ChatGLM 对接使用，用AI来做代码评审。
- 微信公众号的模板配置和API对接使用，运用 API 完成消息触达。
- 通过代码完成 Git 命令使用，检出代码分支，在通过 OpenAI 完成代码评审。整个过程先使用大家常用的流水账方式开发代码，再带着大家重构设计，让代码变得更加清晰。
- 打包 Jar 包，并把相关的组件一起打包，之后让 GitHub Actions 分别通过 Main 函数调用、mvn 构建使用、Jar 下载使用，多种方式学习整个过程（而不是只最终的结果，过程非常重要）。

二、AI对这个项目的介绍

期待作者的接下来的文章，我们顺便问问文心一言对于作者的这个项目的介绍，如下：

AI代码评审组件项目是一个基于GitHub Actions、OpenAI（特别是ChatGLM）、Git/GitHub以及公众号模板消息的综合解决方案。该项目旨在通过自动化流程提高代码评审的效率和质量，从而提升软件开发项目的整体交付质量。以下是对该项目的详细介绍：

项目概述

- **技术栈**：GitHub Actions、OpenAI（ChatGLM）、Git/GitHub、公众号模板消息。
- **目标**：实现从代码提交到通知、Git检出分支变化、使用OpenAI进行代码评审并写入日志，最后通过微信公众号发送消息通知的完整链路。

项目流程

1. **代码提交**：开发者在GitHub上提交代码变更。
2. **Git检出**：GitHub Actions触发，检出最新的代码变更。
3. **代码评审**：利用OpenAI（ChatGLM）进行代码评审，识别潜在的代码问题或改进建议。
4. **写入日志**：将评审结果写入到指定的日志库中，以便后续跟踪和查询。
5. **消息通知**：通过微信公众号发送模板消息，通知相关人员代码评审的结果和变更详情。

项目价值

对于程序员而言，小傅哥的AI代码评审组件项目具有以下价值：

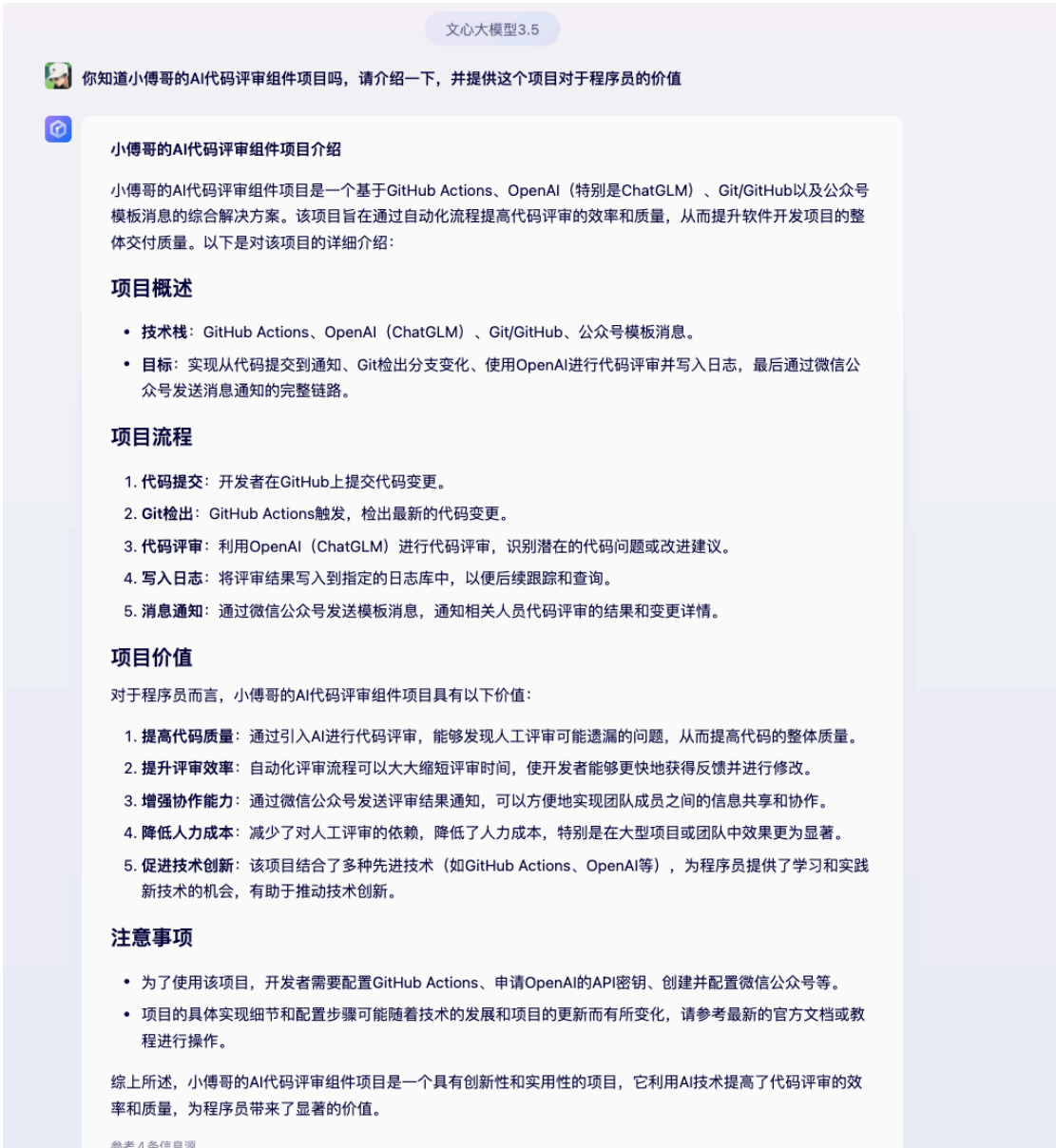
1. **提高代码质量**：通过引入AI进行代码评审，能够发现人工评审可能遗漏的问题，从而提高代码的整体质量。
2. **提升评审效率**：自动化评审流程可以大大缩短评审时间，使开发者能够更快地获得反馈并进行修改。
3. **增强协作能力**：通过微信公众号发送评审结果通知，可以方便地实现团队成员之间的信息共享和协作。
4. **降低人力成本**：减少了对人工评审的依赖，降低了人力成本，特别是在大型项目或团队中效果更为显著。
5. **促进技术创新**：该项目结合了多种先进技术（如GitHub Actions、OpenAI等），为程序员提供了学习和实践新技术的机会，有助于推动技术创新。

注意事项

- 为了使用该项目，开发者需要配置GitHub Actions、申请OpenAI的API密钥、创建并配置微信公众号等。
- 项目的具体实现细节和配置步骤可能随着技术的发展和项目的更新而有所变化，请参考最新的官方文档或教程进行操作。

综上所述，AI代码评审组件项目是一个具有创新性和实用性的项目，它利用AI技术提高了代码评审的效率和质量，为程序员带来了显著的价值。

截图如下：



三、这个项目背后的知识

这个项目中，我们可能会接触到一个新的知识与技能，就是Github Actions，AI大模型给我们的介绍的这个知识如下：

Github Actions 说明

GitHub Actions 是 GitHub 提供的一个持续集成和持续部署（CI/CD）平台，它允许开发者直接在 GitHub 仓库中构建、测试和部署他们的代码。GitHub Actions

通过使用工作流（workflows）来自动化软件开发周期中的任务，这些工作流由一系列预定义的构建和测试任务组成，可以在代码被推送到仓库、创建拉取请求、或者满足其他特定条件时自动触发。

主要特点和优势

1. **集成在 GitHub 中：** GitHub Actions 紧密集成在 GitHub 平台中，使得代码仓库、问题和拉取请求等资源的管理更加统一和便捷。
2. **灵活的自定义能力：** 开发者可以定义自己的工作流，并使用 Docker 容器作为执行环境，这提供了极大的灵活性来运行几乎任何工具或服务。
3. **丰富的预定义动作（Actions）：** GitHub Actions 市场提供了大量的预定义动作，涵盖了从代码检查、测试到部署的各个方面，大大简化了配置过程。

4. **可视化工作流**：GitHub Actions 提供了一个直观的可视化界面，用于查看工作流的执行状态、日志和结果，便于调试和监控。
5. **强大的事件触发机制**：GitHub Actions 支持多种事件触发工作流，包括但不限于代码推送、拉取请求、评论、标签创建等，满足不同场景下的自动化需求。

基本概念

- **工作流 (Workflows)**：定义了一组自动化任务的集合，这些任务在特定事件发生时执行。
- **事件 (Events)**：触发工作流执行的触发器，如代码推送、拉取请求等。
- **作业 (Jobs)**：工作流中的一个执行单元，可以包含多个步骤 (steps)。
- **步骤 (Steps)**：作业中的一个执行指令，可以是一个 shell 命令、一个脚本或一个预定义的动作 (Action)。
- **动作 (Actions)**：可重用的代码包，用于在 GitHub Actions 中执行特定任务。

使用场景

- **自动化测试**：在代码被推送到仓库时自动运行单元测试、集成测试等，确保代码质量。
- **代码检查**：使用静态代码分析工具检查代码中的潜在问题，如代码风格不一致、潜在的bug等。
- **自动化部署**：将代码自动部署到生产环境或测试环境，提高部署效率。
- **自动化通知**：在构建失败、代码合并到主分支等事件发生时，通过邮件、Slack等方式通知相关人员。

入门指南

1. **创建工作流文件**：在仓库的 `.github/workflows` 目录下创建一个 YAML 文件来定义工作流。
2. **配置工作流**：在 YAML 文件中配置工作流的事件、作业、步骤和动作等。
3. **提交和推送**：将工作流文件提交到仓库，并推送到 GitHub。
4. **查看执行结果**：在 GitHub 仓库的 Actions 选项卡中查看工作流的执行状态和结果。

GitHub Actions 提供了一个强大的平台来自动化软件开发周期中的任务，帮助开发者提高开发效率和代码质量。通过不断学习和实践，开发者可以充分利用 GitHub Actions 的优势来优化他们的软件开发流程。

四、学习资料

<https://docs.github.com/zh/actions>