



《OpenAI 代码自动评审组件》- 第2节-学习记录

来自：码农会锁

爱海贼的无处不在

2024年07月21日 12:44

一、正文

在第2节中，作者手把手式的讲解了如何快速的使用Github的Actions的用法，先通过在本地IDEA中创建和初始化完成基本的代码仓库后，然后上传到Github中，并初始化Github的Actions的工作流配置文件，然后设置基本的配置信息和动作，最终完成了一个基于javac命令的Github的Actions的操作。虽然这个操作很少，没啥复杂性，但是对于我们的思考还是很多的。

本节主要是记录下关于Github Actions的相关内容

我们应该在空闲时间深入学习下Github Actions的玩法，正如作者所说，这个功能是我们的一台免费云服务器，我们有些内容可以使用这个功能来帮我们做些事情。比如我们写个程序或脚本，然后通过它自动化的做自己想做的内容等等学习下官方文档还是非常必要的：

<https://docs.github.com/zh/actions/learn-github-actions/understanding-github-actions>

GitHub 文档

Version: Free, Pro, & Team

搜索 GitHub Docs

← 主页

GitHub Actions

快速入门

了解 GitHub Actions

示例

使用工作流

使用作业

管理工作流运行

生成和测试

部署

容器化服务

发布包

管理问题和拉取请求

迁移到 GitHub Actions

监视和故障排除

GitHub 托管的运行程序

自托管运行程序

了解工作流程文件

为帮助您了解如何使用 YAML 语法来创建工作流文件，本节解释介绍示例的每一行：

YAML

Beside

Inline

```
name: learn-github-actions

run-name: ${ github.actor } is learning GitHub Actions

on: [push]

jobs:
```

Optional - The name of the workflow as it will appear in the ' the GitHub repository. If this field is omitted, the name of the will be used instead.

Optional - The name for workflow runs generated from the v will appear in the list of workflow runs on your repository's ". This example uses an expression with the github context to username of the actor that triggered the workflow run. For n information, see "GitHub Actions 的工作流语法."

Specifies the trigger for this workflow. This example uses th so a workflow run is triggered every time someone pushes a repository or merges a pull request. This is triggered by a pu branch; for examples of syntax that runs only on pushes to s branches, paths, or tags, see "GitHub Actions 的工作流语法."

Groups together all the jobs that run in the learn-github-ac workflow.

官方文档是这样介绍的：

GitHub Actions 是一种持续集成和持续交付 (CI/CD) 平台，可用于自动执行生成、测试和部署管道。 您可以创建工作流来构建和测试存储库的每个拉取请求，或将合并的拉取请求部署到生产环境。

GitHub Actions 不仅仅是 DevOps，还允许您在存储库中发生其他事件时运行工作流程。 例如，您可以运行工作流程，以便在有人在您的存储库中创建新问题时自动添加相应的标签。

GitHub 提供 Linux、Windows 和 macOS 虚拟机来运行工作流程，或者您可以在自己的数据中心或云基础架构中托管自己的自托管运行器。

在上周的时候，我分享了2个Github Actions的文档，我觉得也是不错的：



基于GitHubActions的CI实战.pdf



自动构建：如何使用+GitHub+Action+构建镜像？.pdf



8赞，1评论

文档中孔令飞老师说：

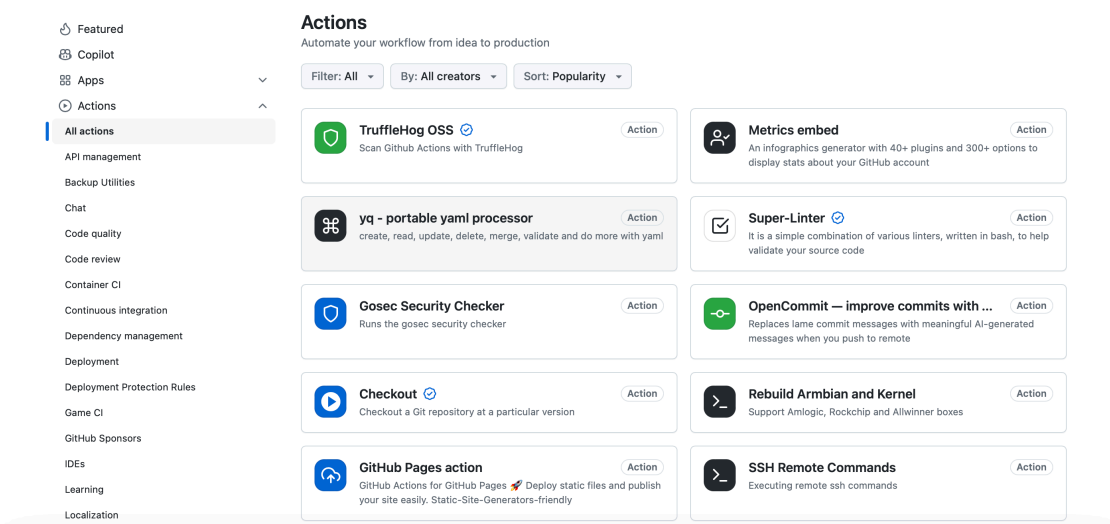
我们用的actions都是可以共享的，我们可以去市场中获取其他人写的进行复用，GitHub允许开发者将这些可共享的Actions上传到GitHub的官方Actions市场，开发者在Actions市场中可以搜索到他人提交的 Actions。另外，还有一个

[awesome actions](#)的仓库，里面也有不少Action可供开发者使用。如果你需要某个 Action，不必自己写复杂的脚本，直接引用他人写好的 Action 即可。整个持续集成过程，就变成了一个 Actions 的组合。Action其实是一个独立的脚本，可以将Action存放在GitHub代码仓库中，通过<userName>/<repoName>的语法引用 Action。例如，actions/checkout@v2表示<https://github.com/actions/checkout>这个仓库，tag是v2。actions/checkout@v2也代表一个 Action，作用是安装 Go编译环境。GitHub 官方的 Actions 都放在 github.com/actions 里面。

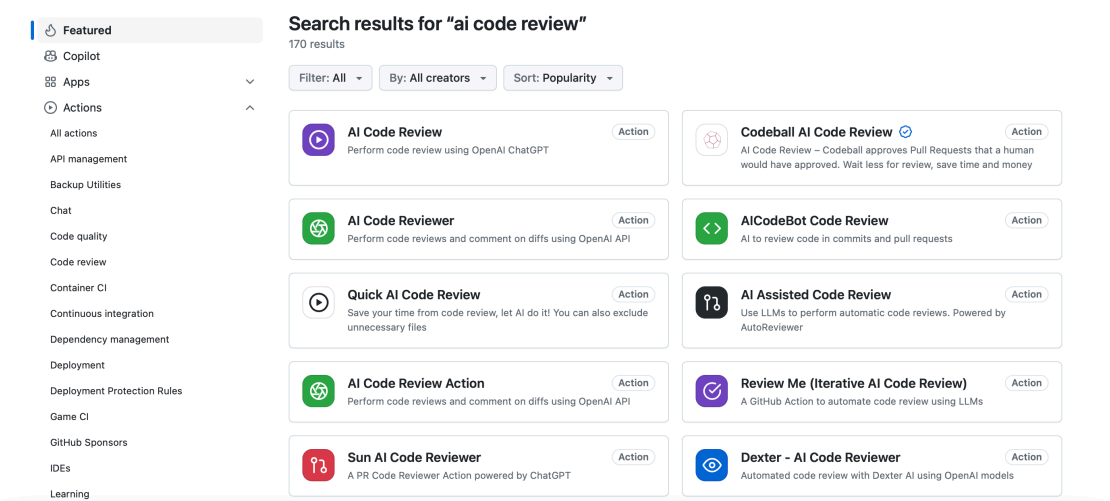
然后我访问了，地址如下：

<https://github.com/marketplace?type=actions>

初步看了一下市场中的共享的actions真不错啊，太棒了，原来这个东西居然这么火，想要的很多玩法，其他网友已经帮助我们实现了类似的，我们完全可以去学习，去使用，去完成我们的自定义的创新想法，提供了各种各样的actions，对于我们的自动化能力真的帮助非常大啊。



而关于AI Code Review的共享action居然高达170个：



那么这个对于我们的思考是什么呢？

我想：**当学习完成了这个小组件项目后，我们是不是可以在包装润色下、在增强创新下、在学习下如何基于前端技术或者后端技术，写写实用的一些Action也不错呢，然后将我们的共享actions给其他人使用呢，当然是没问题的。**

看到有个网友写的Actions信息，居然可以写成这样，结合了调用别人的获取Git代码变更的Actions，又写了一点ollama私有化大模型对接的请求代码，实现了基于私有化大模型进行Github仓库代码评审的例子，真不错。

CodeBlame37 lines (33 loc) · 1.17 KB

RawCopyDownloadEditClose

```
10 steps:
11   - name: Checkout code
12     uses: actions/checkout@v4.1.1
13     with:
14       fetch-depth: 0
15
16   - name: Install Ollama
17     run: |
18       curl -fsSL https://ollama.com/install.sh | sh
19       ollama pull ${ inputs.model }
20     shell: bash
21
22   - name: Get changed files
23     id: changed-files
24     uses: tj-actions/changed-files@v43
25
26   - name: Review code
27     env:
28       GITHUB_TOKEN: ${ github.token }
29     run: |
30       for file in ${ steps.changed-files.outputs.all_changed_files }; do
31         review=$(curl -s http://127.0.0.1:11434/api/generate -d '{"model": "${ inputs.model }"}', "prompt": "Review the following
32           comment="Ollama Code Review for \"${file}\"\\n\\n$review"
33         echo "$comment" >> ollama_review.txt
34       done
35
36       gh pr comment ${ github.event.pull_request.number } --body "${cat ollama_review.txt}"
37     shell: bash
```

二、总结

- 今天学习到了GitHub Actions 的一些术语：
- workflow(工作流程):一个 .yaml 文件对应一个 workflow，也就是一次持续集成。一个 GitHub 仓库可以包含多个 workflow，只要是在 .github/workflow 目录下的 .yaml 文件都会被 GitHub 执行。
 - job(任务):一个 workflow 由一个或多个 job 构成，每个 job 代表一个持续集成任务。
 - step(步骤):每个 job 由多个 step 构成，一步步完成。
 - action(动作):每个 step 可以依次执行一个或多个命令(action)。
 - on:一个 workflow 的触发条件，决定了当前的 workflow 在什么时候被执行。