



1m

# 《OpenAI 代码自动评审组件》- 第8节-学习记录

来自：码农会锁



爱海贼的无处不在

2024年08月20日 23:36

学习系列文章目录：

- 1、《OpenAI 代码自动评审组件》- 第1节-学习记录：[https://articles.zsxq.com/id\\_bpapuj05ur03.html](https://articles.zsxq.com/id_bpapuj05ur03.html)
- 2、《OpenAI 代码自动评审组件》- 第2节-学习记录：[https://articles.zsxq.com/id\\_uokpg7mvt1uc.html](https://articles.zsxq.com/id_uokpg7mvt1uc.html)
- 3、《OpenAI 代码自动评审组件》- 第3节-学习记录：[https://articles.zsxq.com/id\\_bi51vtdhpp36.html](https://articles.zsxq.com/id_bi51vtdhpp36.html)
- 4、《OpenAI 代码自动评审组件》- 第4节-学习记录：[https://articles.zsxq.com/id\\_mw3srnic84z7.html](https://articles.zsxq.com/id_mw3srnic84z7.html)
- 5、《OpenAI 代码自动评审组件》- 第5节-学习记录：[https://articles.zsxq.com/id\\_s55ppt4hbsmr.html](https://articles.zsxq.com/id_s55ppt4hbsmr.html)
- 6、《OpenAI 代码自动评审组件》- 第6节-学习记录：[https://articles.zsxq.com/id\\_pfn3sxd86fa0.html](https://articles.zsxq.com/id_pfn3sxd86fa0.html)
- 7、《OpenAI 代码自动评审组件》- 第7节-学习记录：[https://articles.zsxq.com/id\\_3zaws00xcmj0.html](https://articles.zsxq.com/id_3zaws00xcmj0.html)

## 一、本节内容

本节内容比较简单，即将我们开发好的jar包进行发布，主要讲解了基于提前打包的方式来触发Github Actions的工作流，同时新增加了一个actions的工作流的文件来完成操作，其中将如下内容：

```
- name: Build with Maven
  run: mvn clean install

- name: Copy openai-code-review-sdk JAR
  run: mvn dependency:copy -Dartifact=plus.gaga.middleware:openai-code-review-sdk:1.0 -
  OutputDirectory=./libs
```

替换为了：

```
- name: Create libs directory
  run: mkdir -p ./libs

- name: Download openai-code-review-sdk JAR
  run: wget -O ./libs/openai-code-review-sdk-1.0.jar https://github.com/fuzhengwei/openai-code-
  review-log/releases/download/v1.0/openai-code-review-sdk-1.0.jar
```

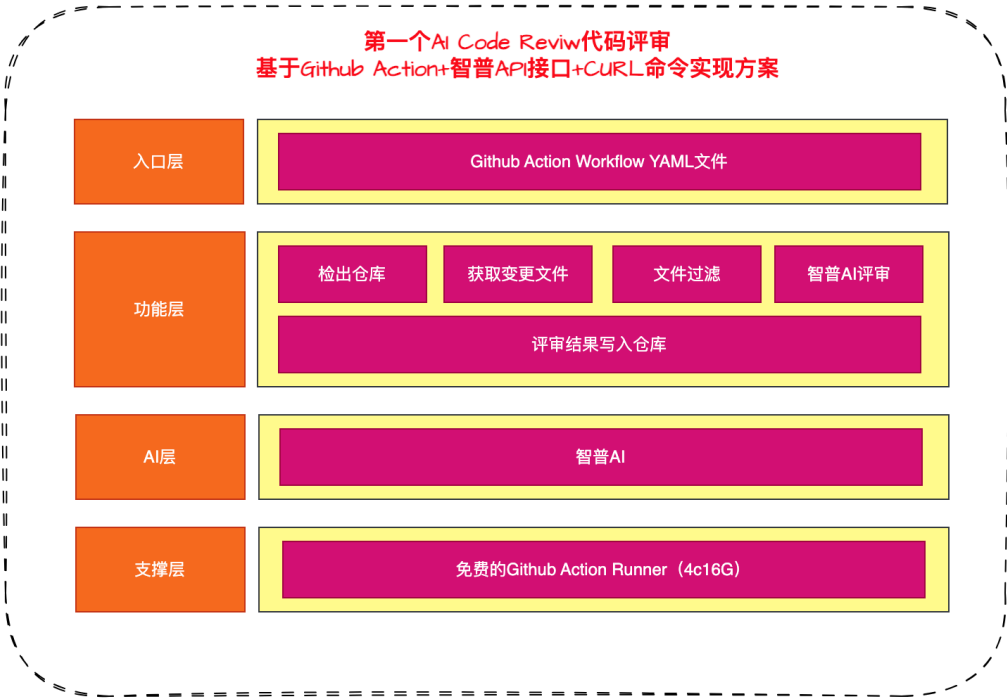
也就说说当我们的Github Actions文件执行时候，从远端拉取核心的Java工具包，然后在Github的环境中执行。这样一个操作。

本文涉及到的内容不多，但是给我们的思考就是：当我们做完了一个这样的组件以后，我们该如何服务最终用户，这个是非常重要的，也是大家需要持续思考的内容。

## 二、总结

至此，我们跟着作者的8节内容就结束了，同时跟着作者的开发，我们也进行了五次代码评审的迭代，让我们来回复一下目前我们做过的几版迭代：

**迭代一：基于Github Action+智普API接口+CURL命令实现方案**

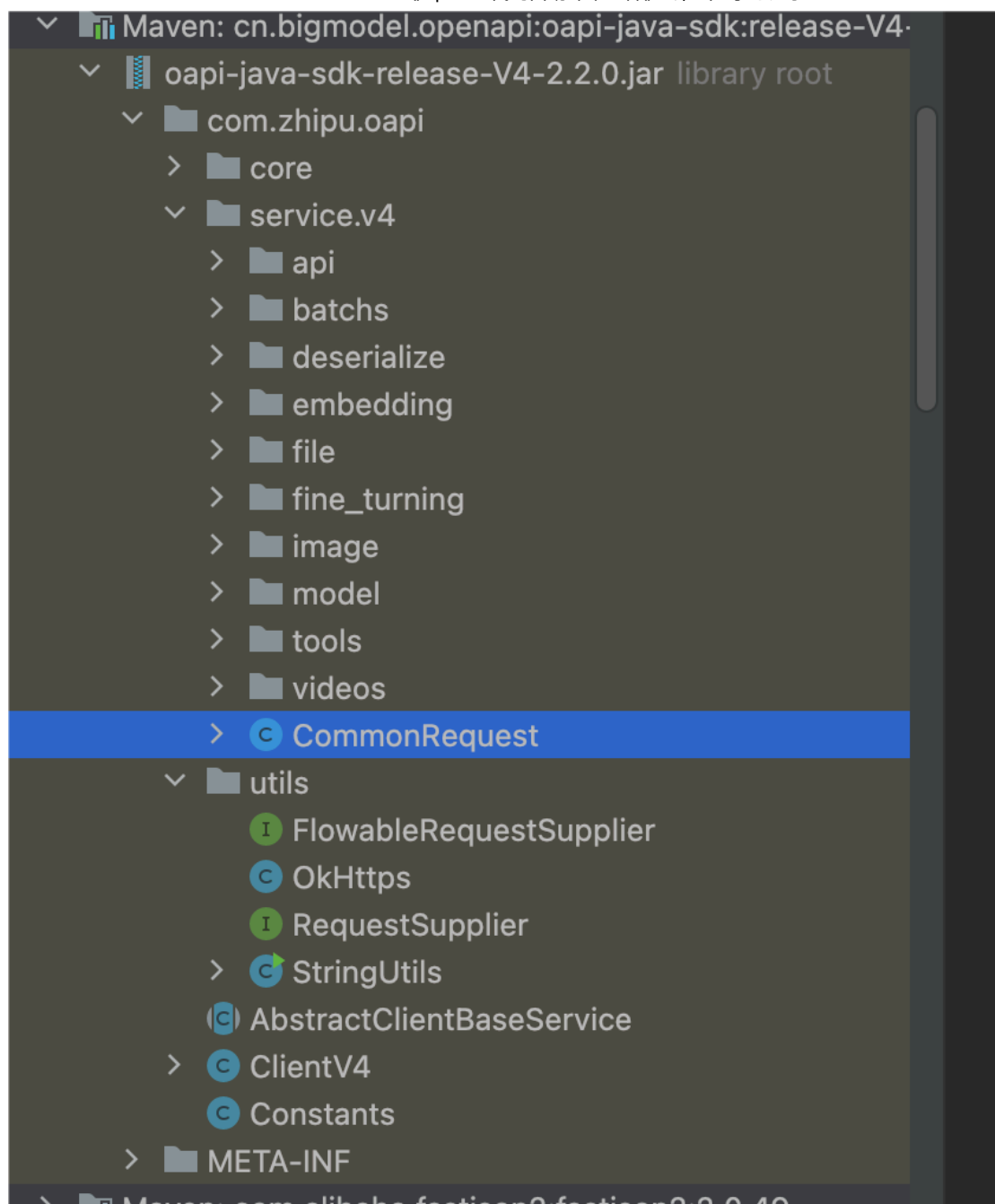


通过迭代一，我们学会了使用CURL命令实现一个文件完成代码评审。甚至不需要Java代码的开发。

迭代二：基于Github Action+智普API接口+Java SDK实现方案



在第二个迭代中，使用了智普的Java SDK的完成了一个基本的代码评审，期间我们应该也自学了智普SDK的核心源码，对于智普SDK里面的那些优秀的抽象模型设计、类结构设计我们也学习了下。例如：

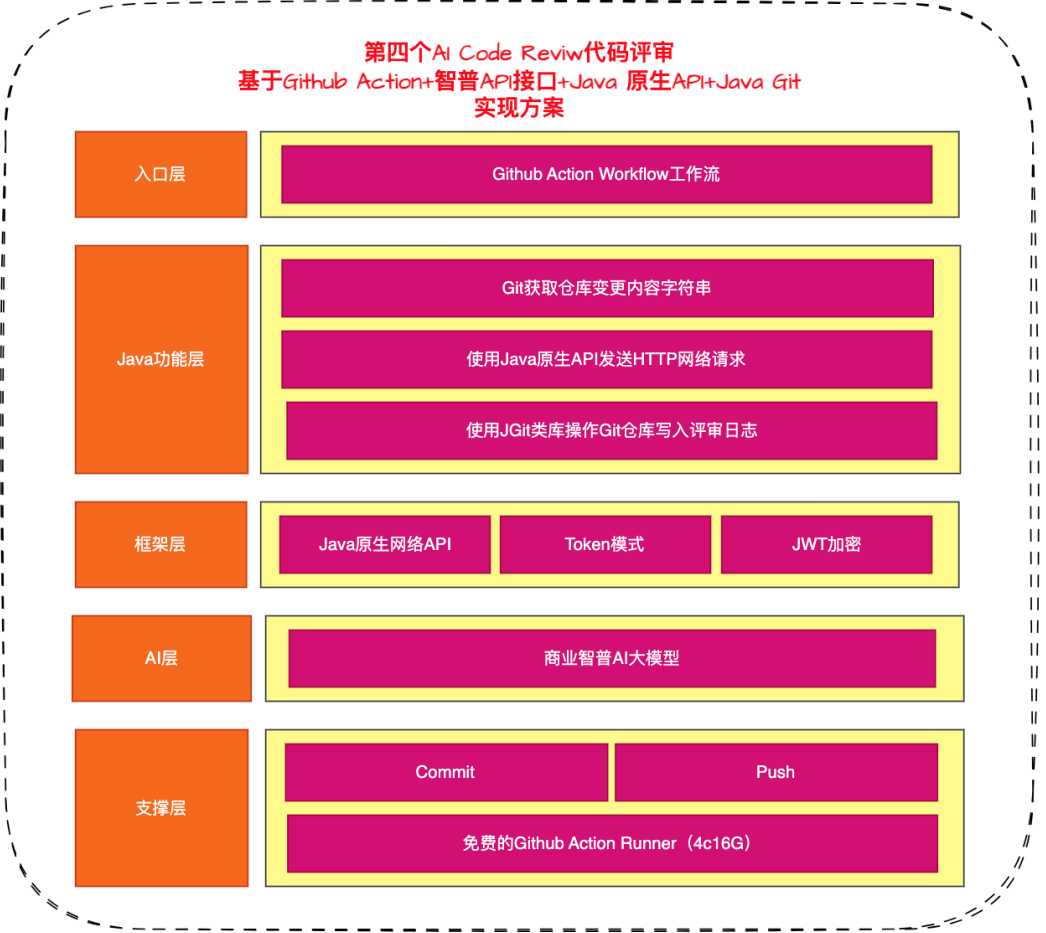


迭代三：基于Github Action+智普API接口+Java 原生API 实现方案



迭代三中，跟着作者使用轻量级的Java 网络API完成代码的评审，同时开始自己封装模型和接口设计。

**迭代四：基于Github Action+智普API接口+Java 原生API+Java Git 实现方案**



在迭代四中，我们跟着作者引进了JGit操作，学会了如何用Java操作Git仓库，以及将我们的评审结果作为文件输出，写入到文件日志中。

**迭代五：基于DDD+Github Action+智普AI+Java 原生API+Java Git 实现方案**



在迭代五中，我们跟着作者完成了基于DDD工程的重构，同时通过模板方法设计模式完成了通用的代码评审的核心业务流程，同时我们也思考了**Pipeline设计模式、生命周期模式、工厂模式、监听者模式**的扩展应用。

经过这样的几个迭代和作者的讲解，以及我们的练习、思考、总结，我们对于一个代码评审的工程架构设计和样子有了初步的一个理解和认识。

至此，我们的OpenAI 代码自动评审组件第一期就结束了，那么OpenAI 代码自动评审组件的二期建设也即将开始，我们先来看看目前初步规划的内容吧：

基于Github的OpenAI 代码自动评审组件的二期的初步想法：

- 1、基于Github实现的对接Commit API的AI的代码评审的设计与实现
- 2、基于Github实现的对接Pull Request API的AI的代码评审的设计与实现
- 3、对接私有化大模型部署工具的Ollama API的设计与实现
- 4、提示词的维护的设计与实现
- 5、SDK的AI代码评审执行结果发送，用于未来的结果反馈，与统计(领导最喜欢东西有指标和度量性的内容)
- 6、AI代码评审中上下文能力的构建支持的设计与实现
- 7、提交文件的过滤的设计与实现
- 8、结合Github Pages操作的设计与实现
- 9、消息通知集成飞书群机器人的消息通知的设计与实现

顺便看看基于Github的OpenAI 代码自动评审组件的三期的初步想法：

- 1、基于Webhook模式的Server端AI代码评审的设计与实现
- 2、基于苍穹外卖的前端Web UI的Server端的维护界面的设计与实现
- 3、基于字节Coze平台的Github的AI代码评审设计与实现
- 4、基于Github某些其他的API辅助代码评审的设计与实现

.....

同时近期计划也会在星球内分享我个人的基于Gitlab平台的AI代码评审项目设计。

结束，即将再见。

