

代码评审助手（CR-Copilot）

功能介绍

一个基于开源大模型的 Code Review 实践，类似一个代码评审助手。

在 Gitlab 中使用 deepseek 或其他模型（ChatGPT等）进行 CodeReview。当你在 GitLab 上创建一个新的 Merge request 时，机器人会自动进行代码审查，审查信息将显示在 MR timeline / file changes 中。

背景：

想法源于在一次 Code Review 时，向 ChatGPT 询问哪种写法代码更优雅得来。当时就想能不能让 AI 帮我们辅助做 Code Review？

部门前后端每天代码提交至少10-20次，但缺乏关注flow流程和代码质量。同时MR审批人制度难以推动建立，虽然提交时经过 单测 + Lint 过滤了一些低级错误，但还有些问题（代码合理性、安全性、MR 相关业务逻辑等）没有过多关注，通常这些工作需要经验更深厚的开发人员来做，且花费大量时间。

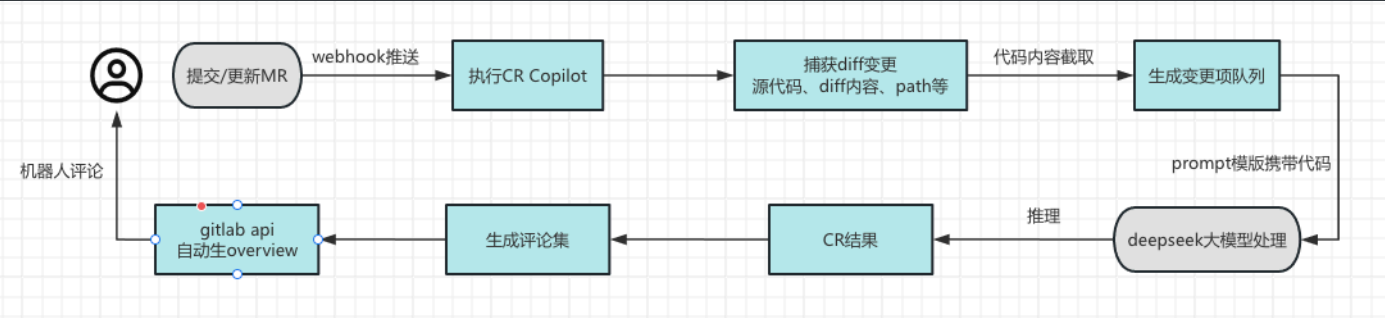
最后可以先经过自动化 CR给出每次变更项的审核和给出修改意见，从实践上降低开发错误发生频次，减少安全漏洞，提高代码健壮性的同时可以完全节省审查流程耗时。

特性：

1. **开箱即用**：基于 Gitlab CI的webhook，仅需url配置完成接入，自动捕获，MR触发时对 MR 进行 自动化CR，生成代码质量监测报告，自动插入discussion。
2. **数据安全**：目前已经接入了公司内部部署的deepseek版本，确保代码 CR 过程仅在内网环境下完成。
3. **无调用次数限制**：使用公司内部部署的本地化deepseek版本，没有其他成本。
4. **评论到变更行**：CR 助手将结果评论到变更代码行上，实现自动化MR。
5. **无编程语言及文件限制**：支持所有编程语言涉及的可读性文件类型。

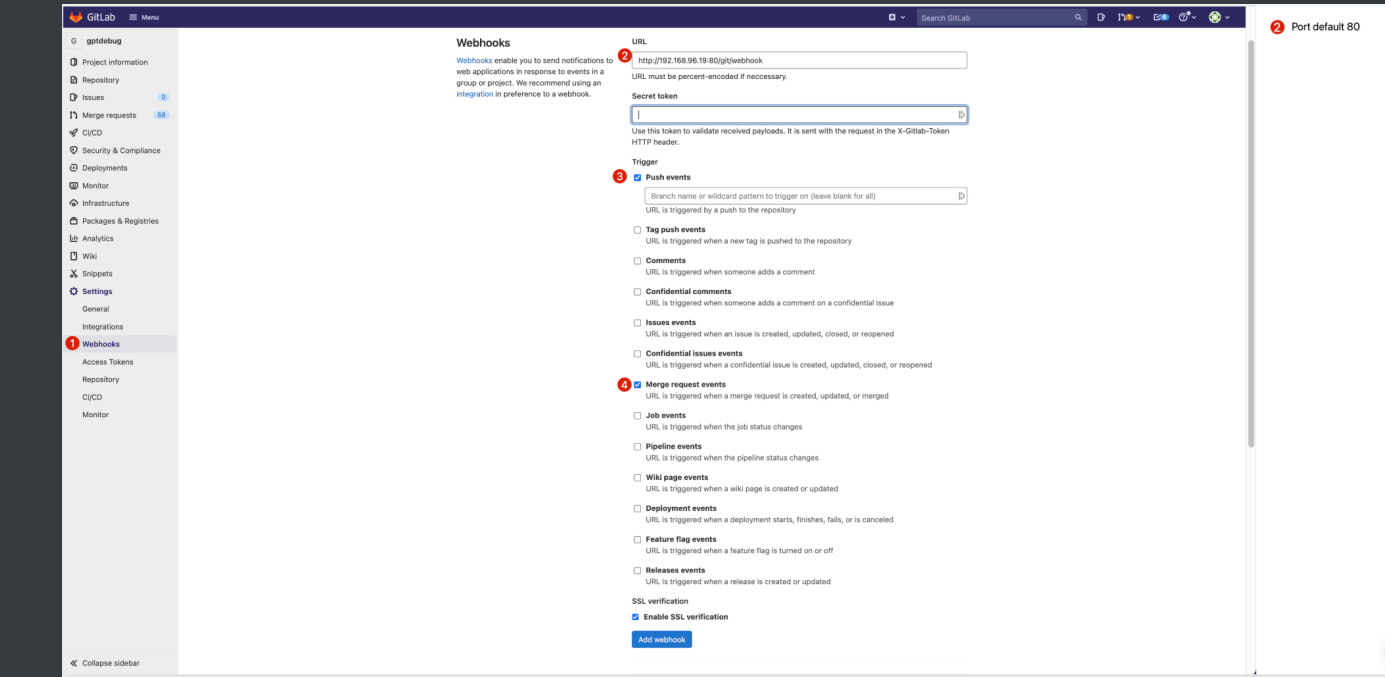
实现思路：

流程图：



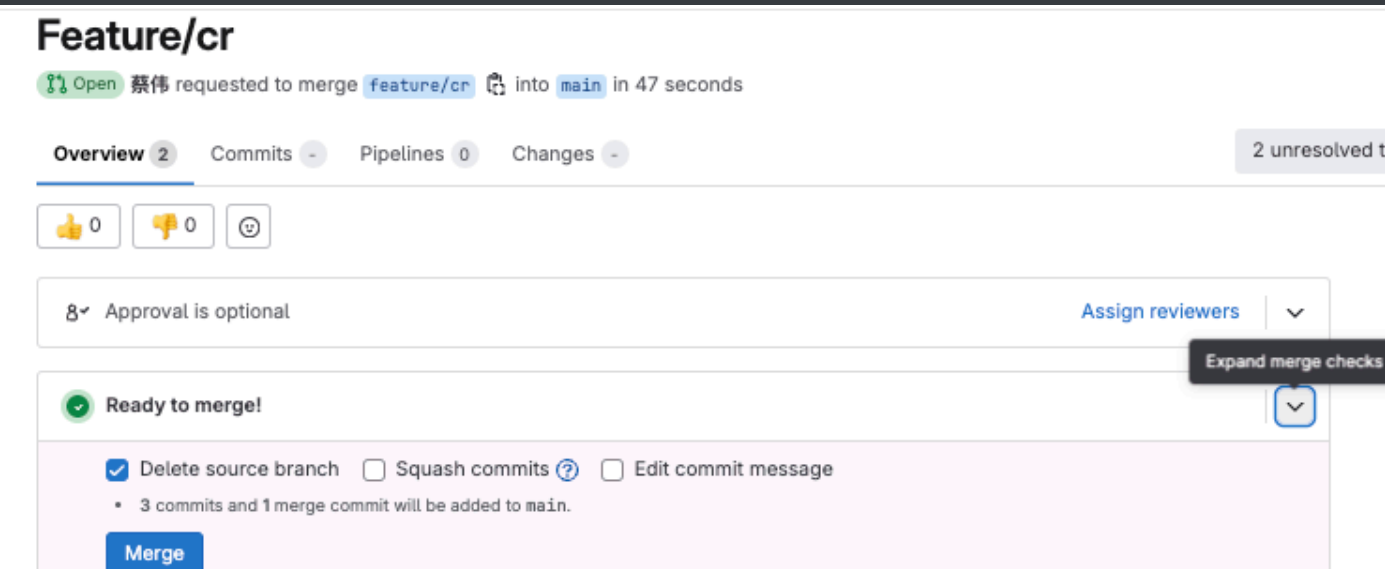
操作步骤:

1. 配置Gitlab webhook



2. 开发代码提交feature分支生成commit

3. 执行Gitlab Merge Request 触发评论



Activity

All activity ▾



cr_test @project_67042648_bot_f9e91e72631e2575308e51b7ea2274d3 started a thread on the diff in 47 seconds

▾ Collapse replies

hello.js 0 → 100644

```
21 + // Draw a circle
22 + ctx.beginPath();
23 + ctx.arc(100, 100, 20, 0, Math.PI * 2, true);
24 + ctx.closePath();
25 + ctx.fillStyle = '#0000ff';
26 +
27 +
28 + // Draw a line
29 + ctx.moveTo(0, 0);
30 + ctx.lineTo(800, 600);
31 + ctx.strokeStyle = '#00ff00';
32 +
33 + // Draw text
34 + ctx.font = '30px Arial';
35 + ctx.fillText('Hello, world!', 10, 50);
36 +
```



cr_test @project_67042648_bot_f9e91e72631e2575308e51b7ea2274d3 · in 47 seconds

Maintainer



好的，我现在要帮用户审查这个GitLab上的代码。首先，我看到用户给了一段JavaScript代码，主要是在HTML文档中创建一个canvas并进行一些绘图操作。

潜在问题方面，第一点是关于绘图顺序的。画线的时候，stroke方法在设置strokeStyle之前调用了lineTo，这样可能导致线条颜色不正确，应该先设置颜色再绘制。第二点是资源管理的问题，canvas元素没有被移除或清理，可能会导致内存泄漏或者性能问题，特别是在需要频繁创建和销毁canvas的情况下。

关于安全漏洞，这段代码看起来并没有明显的XSS或其他注入风险，因为它只是在Canvas上绘图，没有处理用户输入或者动态内容，所以目前看来不存在安全漏洞。

改进建议方面，针对画线的问题，应该调整顺序，在lineTo之前设置strokeStyle。另外，添加canvas的清理机制也很重要，比如在完成后调用remove方法，释放资源。这样可以避免潜在的性能问题和内存泄漏。

1. 潜在问题：绘图顺序问题可能导致部分图形未正确显示。画线操作先于设置 strokeStyle，可能不会应用颜色。
2. 不存在安全漏洞。
3. 改进建议：

```
// Draw a line
ctx.strokeStyle = '#00ff00'; // 移到lineTo之前
ctx.moveTo(0, 0);
ctx.lineTo(800, 600);
ctx.stroke();

// 添加清理机制以释放资源
canvas.remove();
```

Reply...

Resolve thread



cr_test @project_67042648_bot_f9e91e72631e2575308e51b7ea2274d3 started a thread on the diff in 47 seconds