核心 1 浏览器 GitCode 评论插件（开发者）

可选 2 关单模板配置与关单审计（管理员 / 测试）

核心 1 核心功能：浏览器 GitCode 评论插件

1.1 安装与启用

> 装好插件后，在 issue 评论区点击「关单说明」，按表单填写关单要素并提交。

步骤 1

在 Chrome 应用商店在线安装（推荐）

在 Chrome 应用商店 搜索「GitCode Quick」，进入详情页点击「**添加至 Chrome**」，在弹窗中确认「添加扩展程序」。

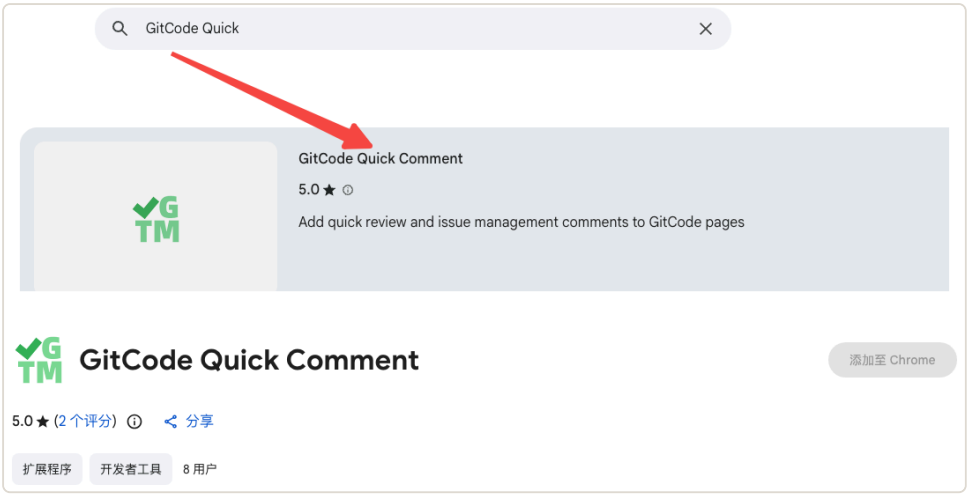


图 1 应用商店搜索「GitCode Quick」找到扩展（上），进入详情页点击「添加至 Chrome」（下）。

步骤 2

本地加载未打包扩展（备选）

开发者或内部分发场景下，打开 `chrome://extensions`，开启「开发者模式」，点击「**加载未打包的扩展程序**」导入解压后的 zip 包；安装完成后确认扩展卡片上的开关处于启用状态。



图 2 点击「加载未打包的扩展程序」，导入解压缩后的 zip 包。

核心 1 核心功能：浏览器 GitCode 评论插件

1.2 issue 关单说明

步骤 3

在 issue 评论区点击「关单说明」

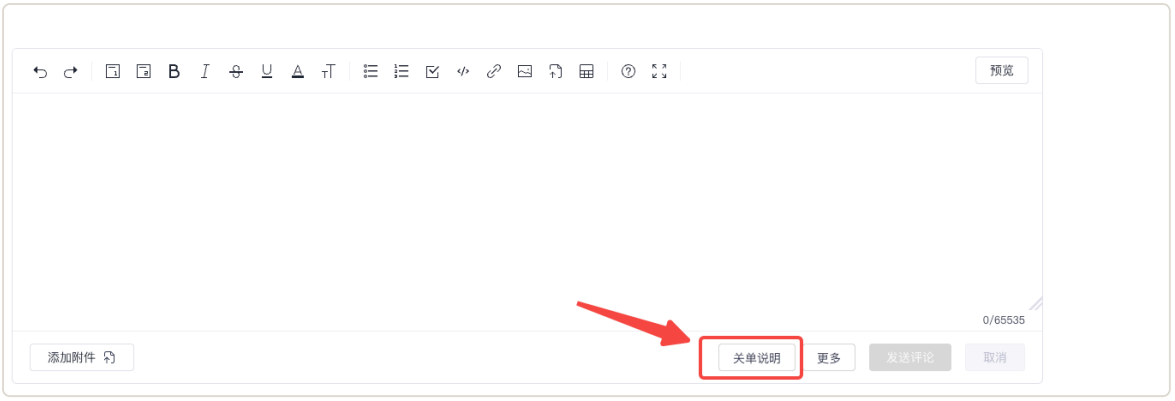


图 3　issue 页面底部评论框中，「关单说明」按钮位于「更多」左侧、「发送评论」右侧，点击即打开关单表单。

步骤 4

按表单填写 7 组要素

表单共 7 组要素：**必填 4 组** —— 02 原因分析、04 用户感知、05 测试回归版本、07 自测验证；**可选 3 组** —— 01 缺陷引入、03 修复方案、06 关联组件，默认折叠，点右侧 `+` 展开填写。



图 4　表单上半部分：01 缺陷引入、02 原因分析、03 修复方案、04 用户感知。



图 5　表单下半部分：05 测试回归版本、06 关联组件、07 自测验证与底部按钮。

核心 1 核心功能：浏览器 GitCode 评论插件

1.2 issue 关单说明（续）

步骤 5

填入评论框并提交

表单底部显示「草稿自动保存在本机，可直接粘贴或拖入图片，检查后手动提交」。点「**填入评论框**」把内容写成 Markdown 填入评论区，人工检查后点「发送评论」提交；最后添加 **resolved** 标签 —— 标签一加上，若启用了关单审计就会触发审计。

提交前对照：7 组要素

编号	要素	填写状态	表单中的字段
01	缺陷引入	可选 · 展开填写	默认折叠，点击 + 展开
02	原因分析	**必填**	引入缺陷的原因：触发条件、根因、影响范围、关键代码 / 日志
03	修复方案	可选 · 展开填写	默认折叠，点击 + 展开
04	用户感知	**必填**	是否引入用户可感知变更：是 / 否
05	测试回归版本	**必填**	回归包类型：daily / tag；包名 / 版本号；回归包链接
06	关联组件	可选 · 展开填写	默认折叠，点击 + 展开
07	自测验证	**必填**	自测截图与结论：自测环境、自测步骤 / 命令、自测结果

PR 评论快捷键（可选）

同一插件在 PR 页面的评论框上还提供一组快捷键，用于快速完成评审意见与流水线操作，按需使用即可。

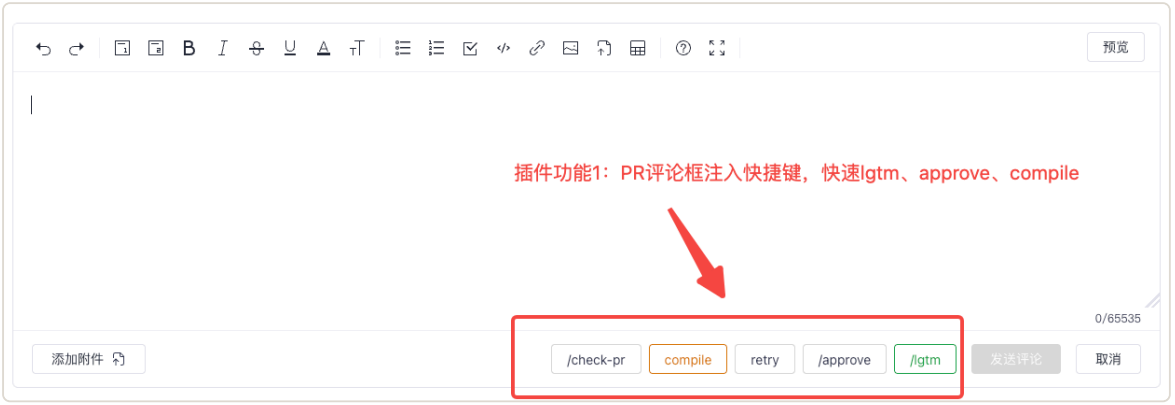


图 6　PR 评论框下方的快捷键：`/check-pr`、`compile`、`retry`、`/approve`、`/lgtm`。

可选 2 可选功能：关单模板配置与关单审计

2.1 关单模板配置（管理员）

> 【可选】关单模板决定表单长什么样、审计查哪些字段。管理员在仓库根目录的 `.msbot/` 下维护机器人行为与关单模板；关单模板遵循 Issue Forms 的 YAML 文件格式，可按需自定义。不启用这一项时，插件使用默认表单。

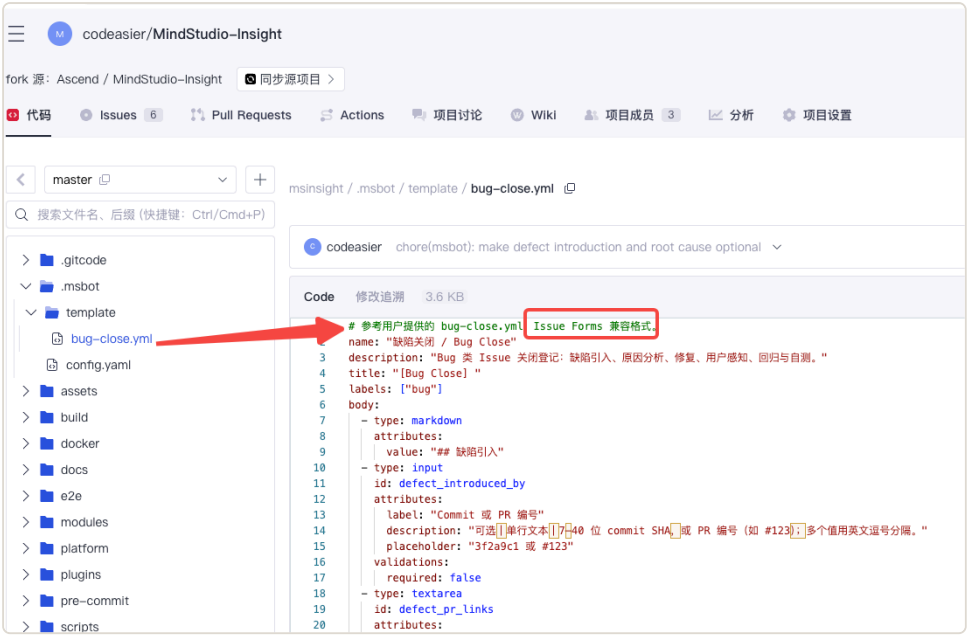


图 7　仓内的关单模板 `bug-close.yml`（左侧文件树中与 `config.yaml` 同级）与其模板内容，文件头注明为「Issue Forms 兼容格式」。

关单模板的写法

```yaml
# 参考用户提供的 bug-close.yml — Issue Forms 兼容格式
name: "缺陷关闭 / Bug Close"
description: "Bug 类 Issue 关闭登记：缺陷引入、原因分析、修复、用户感知、回归与自测。"
title: "[Bug Close]"
labels: ["bug"]
body:
  - type: markdown
    attributes:
      value: "## 缺陷引入"
  - type: input
    id: defect_introduced_by
    attributes:
      label: "Commit 或 PR 编号"
```

模板的 `body` 列表按 Issue Forms 的写法声明字段：`markdown` 输出小节标题，`input` / `textarea` 声明输入项，`id` 与 `label` 决定字段名。插件按这份模板渲染表单，关单审计也按同一份模板校验必填项。

可选 2 可选功能：关单模板配置与关单审计

2.2 关单审计（管理员 / 测试）

【可选】启用关单审计需要两件事：`.msbot/` 下的模板与机器人行为配置就绪，加上仓库把 Issue 事件推给机器人的 Webhook。飞书机器人的 Webhook 链接由管理员 / 测试提供。

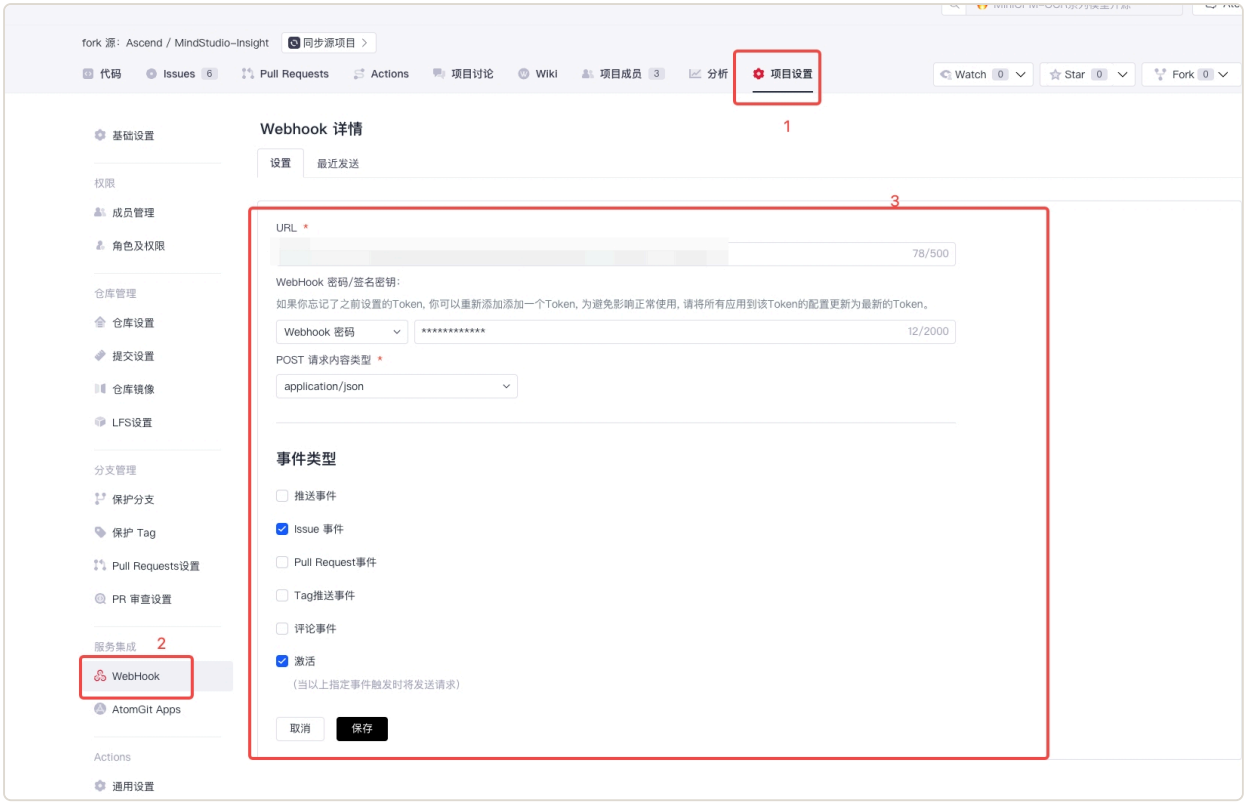


图 8　项目设置 → 服务集成 → Webhook：填入 URL 与签名密钥，勾选「Issue 事件」并激活。

启用步骤

- 进入仓库「项目设置」→「服务集成」→「Webhook」
- URL 填入管理员 / 测试提供的飞书机器人 Webhook 地址
- 按需设置 WebHook 密码 / 签名密钥
- POST 请求内容类型选 `application/json`
- 事件类型勾选 **Issue 事件**，再勾选「激活」并保存

安全提示　飞书机器人 Webhook 地址与签名密钥属于敏感配置，只在项目设置中保存，不要写入仓库文件或提交记录。

可选 2 可选功能：关单模板配置与关单审计

2.2 关单审计（管理员 / 测试）

> 【可选】issue 被添加 resolved 标签时，机器人触发关单审计，按模板逐项校验必填要素。若开发者的关单说明要素不完整，机器人会强制删除 resolved 标签，并推送飞书机器人通知。



图 9　审计不通过时，机器人移除刚被添加的 resolved 标签，issue 停留在未关闭状态。

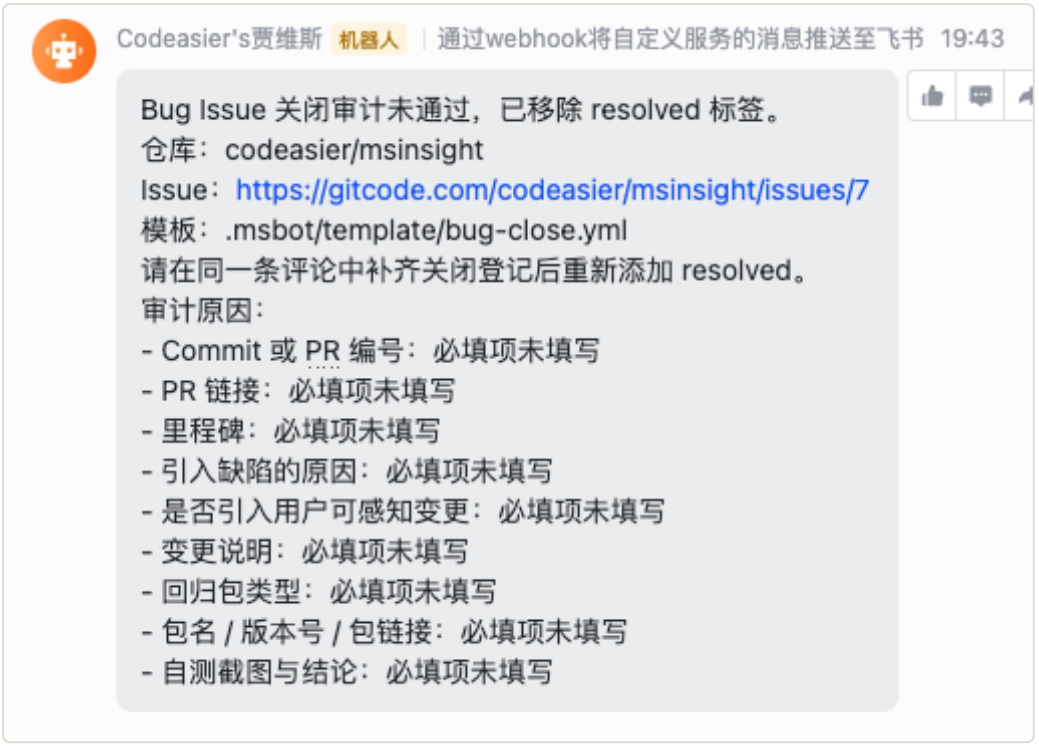


图 10　飞书群内的审计通知。

通知里包含审计结论（未通过、已移除 resolved 标签）、仓库与 issue 链接、所用模板路径、整改提示（补充关闭登记后重新添加 resolved），以及逐条列出的未填写必填项。

> **前提**　飞书通知依赖管理员 / 测试提供的机器人 Webhook 链接。该链接属于敏感配置，只在仓库项目设置中保存，不写入代码库。