



链滴

基于 Gitlab 进行开发团队管理的尝试——0 1. 周报

作者: [crick77](#)

原文链接: <https://ld246.com/article/1574181599600>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

相关代码提交至Github -> <https://github.com/wangyuheng/keeper> 欢迎提交issue

周报的目的在于团队间的沟通交流，总结每周的得失经验。不要流于形式，成为流水账。

What

通过Gitlab issue管理周报，每周定时自动排版并发送周报邮件

Why

痛点

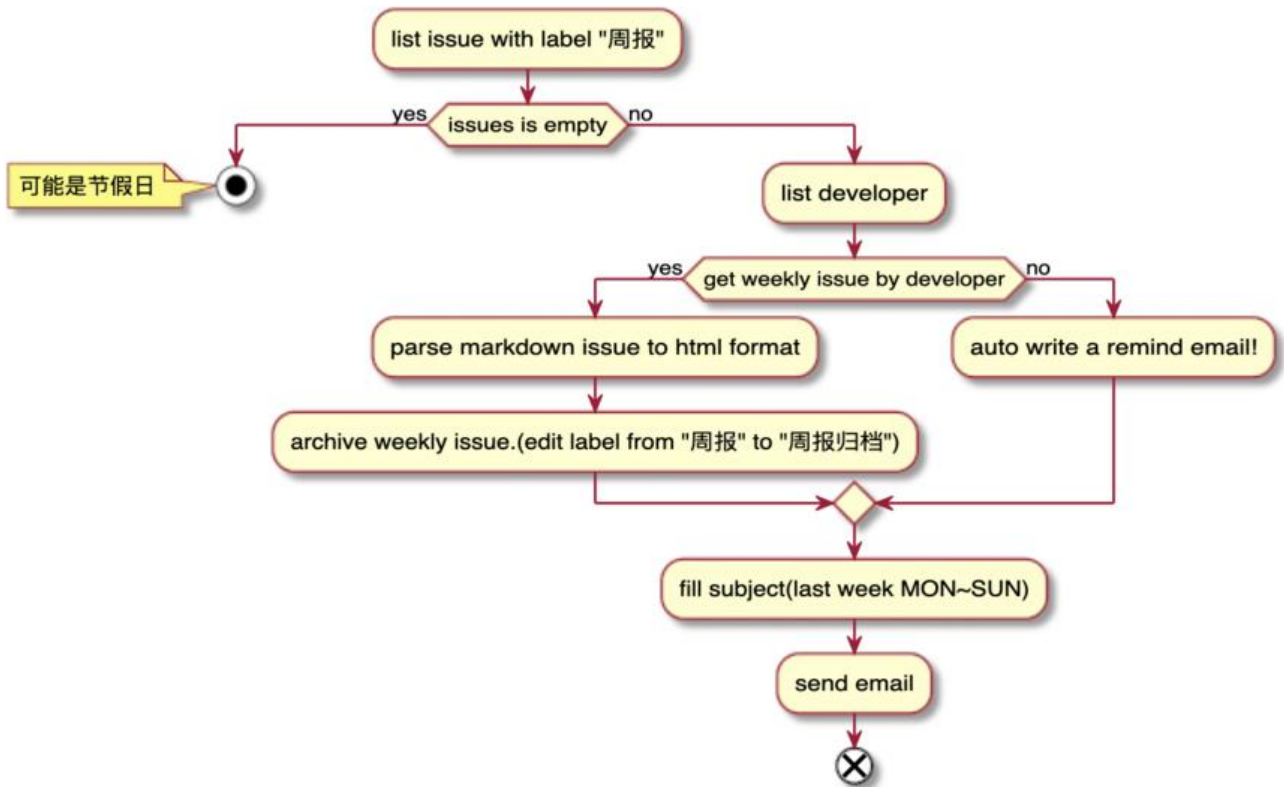
1. 排版格式繁琐，outlook对程序员不算友好
2. 不利于归档管理
3. 督促团队写周报 -> 当然这个主要原因在于leader。如果你写的周报没人关心，没人过问，那么为什么还要写？

优势

1. **markdown**语法 + 统一管理主题、收件人等繁琐信息，让编写者专注于周报内容本身
2. **issue**模板统一风格，提升周报质量&价值
3. 归档管理周报信息，通过 **issue**本身的功能，可以进行人员索引、labels管理等
4. 每周定时扫描并发送邮件。如果扫描时没有发现对应developer的 **周报issue**，那么就广播一封r mind邮件。

三体中逻辑通过摇篮系统，成为了执剑人(Damocleser)，建立黑暗森林威慑。

整体流程如下：



How

对周报编写方来说，只需要在issue中选择**周报**模板并添加对应**label**即可。

Gitlab 设置

1. 添加 **label**，命名为**周报**
2. 在项目 `.gitlab/issue_templates` 目录下新建**周报.md**作为模板，可以在新建**issue**时被选择。

上周最重要的“3”件事

1.

本周最重要的“3”个目标

1.

遇到的问题和感受

1.

扫描job

基于kotlin实现一版定时任务

dependents:

1. h2 -> 内存数据库, 用于存储 **developer**信息
2. commonmark -> 用于 **markdown**转换为**html**格式
3. okhttp & fastjson -> **gitlab** api调用及解析

核心代码如下:

```
@Scheduled(cron = "\${weekly.cron}")
fun checkWeeklyIssue() {
    log.info("start weekly job ! weeklyProjectList:{}, weeklyProjectList)
    val nameMapWeeklyIssue = weeklyProjectList.flatMap {
        gitlabClient.listOpenProjectIssue(it)
    }.filter { issue ->
        issue.getJSONArray("labels").contains(WEEKLY_LABEL)
    }.filter { issue ->
        !issue.getJSONArray("labels").contains(WEEKLY_ARCHIVE_LABEL)
    }.map { issue ->
        val name = issue.getJSONObject("author").getString("name")
        name to WeeklyIssue(issue.getIntValue("project_id"), issue.getIntValue("iid"), name, issue
        getString("description"))
    }.toMap()
    log.info("map gitlab issue data! nameMapWeeklyIssue:{}, nameMapWeeklyIssue)
    // 如果一个记录也没有, 可能是节假日导致。此时不发生提醒
    if (nameMapWeeklyIssue.isNotEmpty()) {
        developerRepository.findAll().forEach {
            val weeklyIssue = nameMapWeeklyIssue[it.name]
            this.sendEmail(weeklyIssue, it)
        }
    }
}

private fun sendEmail(weeklyIssue: WeeklyIssue?, developer: Developer) {
    if (weeklyIssue == null) {
        log.info("send a remind email! name:{}, developer.name)
        emailClient.send(fillSubject(developer.name), "I am ${developer.name}. I do not write a
        eekly! Please remind me when you see me!", developer.receivers)
    } else {
        log.info("send a weekly email! name:{}, developer.name)
        val content = renderer.render(parser.parse(weeklyIssue.description))
        emailClient.send(fillSubject(developer.name), content, developer.receivers)
        gitlabClient.editIssueLabels(weeklyIssue.projectId, weeklyIssue.issueId, WEEKLY_ARCHIVE
        LABEL, true)
    }
}

private fun fillSubject(name: String): String {
    val lastWeek = LocalDate.now().minusWeeks(1)
    return "[周报][XX工程][$name][${lastWeek.with(DayOfWeek.MONDAY)}~${lastWeek.with(Da
    OfWeek.SUNDAY)}]"
}
}
```

问题

1. 节假日如何处理? 如果一包含 **周报**label的issue都没有, 认为是节假日, 跳过后续处理。非周一

送, 由Leader手动触发

2. SLA? 不保障, 手动触发即可