



链滴

基于 Gitlab 进行开发团队管理的尝试——0 3.Dingding 消息 @人

作者: [crick77](#)

原文链接: <https://ld246.com/article/1580645489219>

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

What

监听Gitlab Hook事件, 紧急消息通过钉钉机器人发送至钉钉群组, 并@(提醒)相关方. 同时自动流转issue的pipeline.

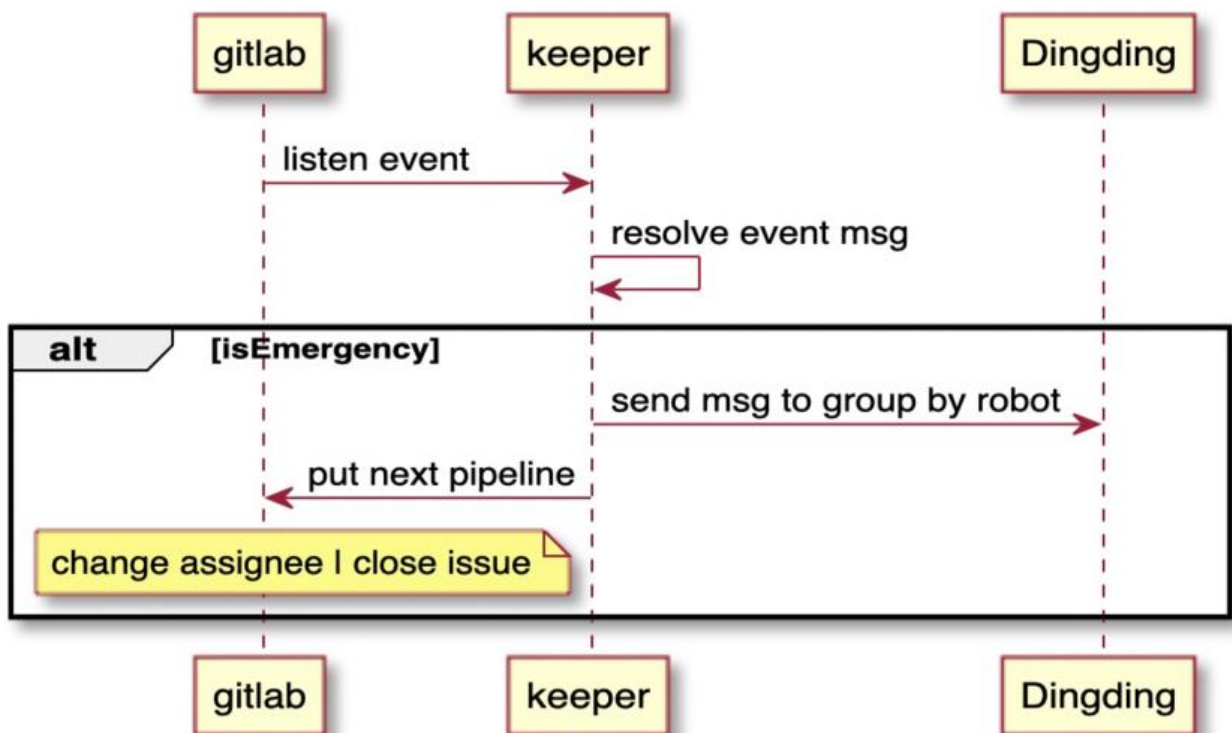
Why

1. 紧急事项实时提醒到人
2. 定制pipeline & 自动流转

为什么不通过钉钉自带的gitlab机器人 或者 gitlab notify email?

1. 发送的消息过多, 会忽略有意义的信息
2. 不能 @ 到相关的人, 达不到提醒的作用

How

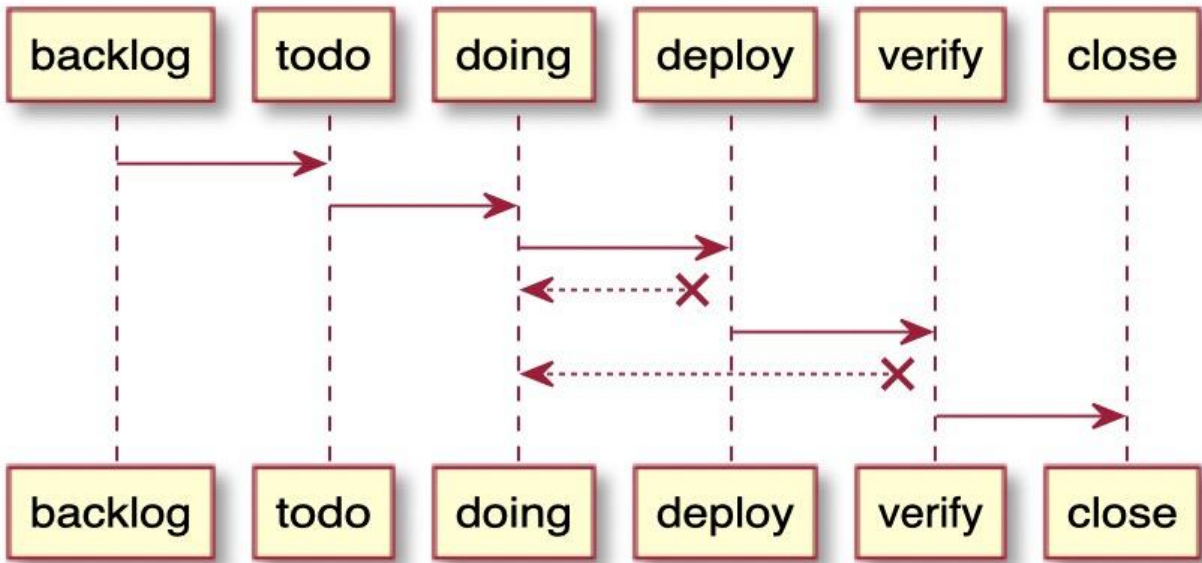


pipeline

根据团队状况制定pipeline, 基于事件自动流转. 比如:

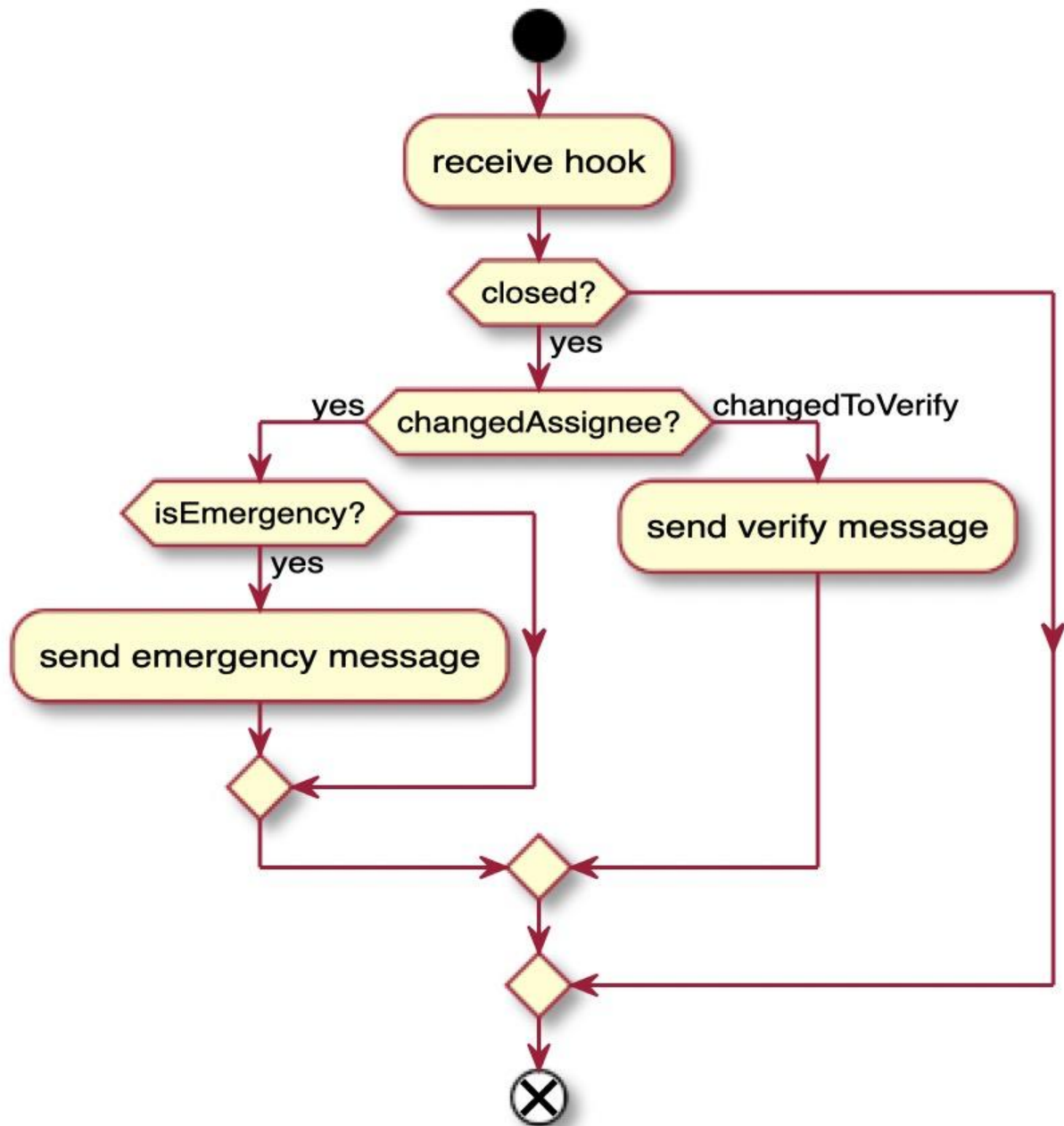
1. 新建 :WIP MR时流转至doing

- 2. CICD流转至deploy
- 3. 发布成功后流转至verify并修改assignee为author



Emergency issue

通过label判断是否为紧要消息, 比如 Bug、P0.



core code

```

private fun handlerIssue(body: JSONObject) {
    if (GitlabHookBodyHelper.isClose(body)) {
        //已关闭的issue不做处理
        return
    }
    val changedAssignee: GitlabHookBodyHelper.Assignee? = getChangedAssignee(body)
    log.info("handler issue changedAssignee: $changedAssignee")
    val labelTitles: MutableSet<String> = GitlabHookBodyHelper.listLabelTitle(body)
    log.info("handler issue labels: $labelTitles")
    // 是否变更assignee
    if (changedAssignee != null) {
        val developer = developerRepository.findByUsername(changedAssignee.username!!)
    }
}
  
```

```
// 为开发人员 && label为todo且不在进行中
if (developer != null && labelTitles.isTodo() && !labelTitles.isProcessing()) {
    when {
        labelTitles.isBug() -> sendBugMsg(body, developer.mobile)
        labelTitles.isPO() -> sendEmergencyMsg(body, developer.mobile)
        else -> log.info("ignore issue change! labelTitles -> $labelTitles")
    }
}
} else {
    if (GitlabHookBodyHelper.isFirstChangeToVerifyLabel(body)) {
        val author = getAuthor(body)
        val assigneeUsername = editAssignee(body, author)
        val mobile = developerRepository.findByUsername(assigneeUsername)?.mobile
        if (null != mobile) {
            sendVerifyMsg(body, author, mobile)
        }
    }
}
}
```

小技巧

非内网部署gitlab如何开发调试

ngrok 实现内网穿透

```
brew cask install ngrok
```

验证阶段的issue关闭不及时

先自动close issue, 并dingding提醒. 如果有问题可以reopen并自动流转至Doing.