



B3log 开源

# Lute PDF - Markdown 生成 PDF

作者: [88250](#)

原文链接: <https://github.com/88250/lute-pdf>

来源网站: [GitHub](#)

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

## Solo 评论组件

在 v4.3.0 发布后，因为评论系统的问题给大家造成了很多不便，非常抱歉。通过大家的[讨论](#)，我们决定在 v4.3.1 中解决该问题。

解决方案是引入 [Gitalk](#) 作为默认评论组件，它会将评论数据保存到用户自己的 GitHub 仓库 issues。在集成 Gitalk 后，Solo 评论功能方面总体的业务逻辑如下：

- 皮肤的文章模板中内置 `<div id="gitalk-container"></div>`
- 用户通过配置页头、页脚集成 Gitalk 评论组件
- 文章推送到社区后，系统会自动外加 VComment 评论组件

也就是说，最多会出现两块评论区：Gitalk 和 VComment，这样可以方便不同访客评论文章：

- 没有黑客派账号的访客可通过 GitHub 登录评论，不必注册黑客派
- 有黑客派账号的访客能可通过社区登录评论

由访客自由选择，在最大程度上方便访客和博主进行交互，于此同时，我们也能通过 GitHub 或者黑派对恶意评论账号进行追溯。

## 配置方式

1. 创建 GitHub OAuth 应用 <https://github.com/settings/applications/new>

- **Application name** 建议填成 `solo-blog-gitalk`
- **Homepage URL** 和 **Authorization callback URL** 都填成线上的博客 URL

2. 创建一个公开仓库 `solo-blog` 用于保存评论数据（该仓库也可以用来备份文章，详见 [solo-blog 仓库同步功能回来了！](#)）

3. 在 Solo 后台的偏好设定 - 信息配置中配置页头和页脚

页头：

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/gitalk@1/dist/gitalk.css">
<script src="https://cdn.jsdelivr.net/npm/gitalk@1/dist/gitalk.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/js-md5/build/md5.min.js"></script>
```

页脚：

```
<script>
document.addEventListener("DOMContentLoaded", function (event) {
  if (Label.articleId) {
    const gitalk = new Gitalk({
      clientId: 'OAuth 应用那里可以找到',
      clientSecret: 'OAuth 应用那里可以找到',
      repo: 'solo-blog',
      owner: '你的 GitHub 用户名',
      admin: ['你的 GitHub 用户名'],
      id: Label.articleId,
      distractionFreeMode: false
    })
  }
})
```

```
    gitalk.render('gitalk-container')
  }
})
</script>
```

4. 配置完成后后，访问文章就可以看到集成 Gitalk 后的效果了，博主通过集成的 Gitalk 登录后，solo blog 仓库上会自动创建和该文章对应的一个 issue 用来保存评论数据

## 细节事项

- Gitalk 配置项 `id` 我们使用的是文章 id，Gitalk 创建 issue 时会使用该 id 作为 issue 的 Label，用 id 的好处是 id 唯一，且长度可控，减少出错的可能性；另外，只要 id 不变，即使迁移博客后也以关联到之前的评论，不会丢失数据
- Gitalk 可能会因为网络问题（比如 GFW）导致不够稳定，如果使用过程中出现报错，请先到 [Gitalk Issues](#) 上寻找解决方案