

commit message 应该是清晰明了，说明本次提交的目的。这样才有利于版本控制。

格式化的提交信息有诸多好处，本文将整理一套自己用的提交信息格式，并在以后的项目中应用起来。

格式

提交信息包括三个部分：Header，Body 和 Footer。

其中，Header 是必需的，Body 和 Footer 可以省略。

Header

Header部分只有一行，包括两个字段：type（必需）和subject（必需）。

:

type

type用于说明 commit 的类别，可以使用如下类别：

- feat：新功能（feature）
- fix：修补bug
- doc：文档（documentation）
- style：格式（不影响代码运行的变动）
- refactor：重构（即不是新增功能，也不是修改bug的代码变动）
- test：增加测试
- chore：构建过程或辅助工具的变动

subject

subject是 commit 目的的简短描述。

- 以动词开头，使用第一人称现在时，比如change，而不是changed或changes
- 第一个字母小写
- 结尾不加句号（。）

Body

Body 部分是对本次 commit 的详细描述，可以分成多行。下面是一个范例。

```
More detailed explanatory text, if necessary.  Wrap it to about 72
characters or so.  Further paragraphs come after blank lines.-  Bullet
points are okay, too-  Use a hanging indent
```

注意：应该说明代码变动的动机，以及与以前行为的对比。

Footer

Footer 部分只用于两种情况：

- 关联 Issue
- 关闭 Issue

关联 Issue

本次提交如果和摸个issue有关系则需要写上这个，格式如下：

```
Issue #1, #2, #3
```

关闭 Issue

如果当前提交信息解决了摸个issue，那么可以在 Footer 部分关闭这个 issue，关闭的格式如下：

```
Close #1, #2, #3
```

例子

说了半天不给个例子都是瞎扯淡，下面是一个完整的例子：

```
feat: 添加了分享功能给每篇博文添加了分享功能- 添加分享到微博功能- 添加分享到微信功
能- 添加分享到朋友圈功能Issue #1, #2Close #1
```

上面的提交信息应该能够自解释自己的意思了。

总结

上面的规范基本够自己用了，我会根据自己的需要，扩展这份规范，你也可以直接使用或者制定自己的规范，快来让我们的提交会说话吧。

参考资料

- [Commit message 和 Change log 编写指南](#)