



# Git 规范

## 一、为什么需要制定提交规范？

在团队协作开发时，每个人提交代码时都会写 commit message。

每个人都有自己的书写风格，翻看我们组的 git log, 可以说是五花八门，十分不利于阅读和维护。

一般来说，大厂都有一套的自己的提交规范，尤其是在一些大型开源项目中，commit message 都是十分一致的。

因此，我们需要制定统一标准，促使团队形成一致的代码提交风格，更好的提高工作效率，成为一名有追求的工程师。

## 二、业界通用的 git 提交规范有哪些？

### 1. commitizen

[AngularJS](#) 在 github 上的提交记录被业内许多人认可，逐渐被大家引用。

格式：

代码块

```
1 type(scope) : subject
```

(1) type（必须）：commit 的类别，只允许使用下面几个标识：

- feat : 新功能
- fix : 修复bug
- docs : 文档改变
- style : 代码格式改变
- refactor : 某个已有功能重构
- perf : 性能优化
- test : 增加测试
- build : 改变了build工具 如 grunt换成了 npm
- revert : 撤销上一次的 commit
- chore : 构建过程或辅助工具的变动

(2) scope (可选) : 用于说明 commit 影响的范围, 比如数据层、控制层、视图层等等, 视项目不同而不同。

(3) subject (必须) : commit 的简短描述, 不超过50个字符。

commitizen 是一个撰写合格 Commit message 的工具, 遵循 Angular 的提交规范。

## 安装:

全局安装 commitizen

代码块

```
1 npm install -g commitizen
```

进入项目文件夹, 运行如下命令:

代码块

```
1 commitizen init cz-conventional-changelog --save --save-exact
```

## 使用:

用 `git cz` 命令取代 `git commit` (先使用git add), 这时会出现如下选项:

(1) 选择 type

! Cannot import this image, please save the image from original file and upload manually.

## (2) 填写 scope (选填)

代码块

```
1 ? What is the scope of this change (e.g. component or file name)? (press enter to skip)
2 core
```

## (3) 填写 subject

代码块

```
1 ? Write a short, imperative tense description of the change:
2 set a to b
```

完成，运行 `git log` 命令，查看我们刚才提交的 commit message，如下：

代码块

```
1 fix(core): set a to b
```

### 优点：

- 符合业内标准（许多项目使用 AngularJS 的 commit 规范）
- 提交过程更加规范（使用 commitizen 规范工具，风格统一）
- 能够生成风格统一的 commit log (type(scope):subject)

### 缺点：

- 需要安装 commitizen 工具包，使项目更大、更重了（适合大型开源项目）
- 提交过程受约束较大
- 有一定的学习成本

## 2. 设置 git commit 模板

步骤如下：

### (1) 建立模板文件

在项目中建立 `.git_template` 文件，内容可以自定义：

代码块

```
1 type:
2 scope:
3 subject:
```

## (2) 设置模板

运行如下命令：

代码块

```
1 git config commit.template .git_template // 当前项目
2 <!-- git config commit.template .git_template // 全局设置 -->
```

## (3) 提交代码

先使用 `git add` 添加代码

使用 `git commit` 按照模板填写

最后 `git push` 推送到远端

优点：

- 规则可配置，更自由
- 配置方式简洁（只需添加配置文件）

缺点：

- 便利性差，每次都要用 vim 编辑器填写模板
- 易出错，没有可靠的校验方式

## 三、制定适合我们的 git commit 提交规范

第二章中提到的两种业内普遍使用的规范，都不完全适合我们。

第一种方式适合大型开源项目，我们如果也照搬会比较麻烦，但我们可以借鉴 `type(scope): subject` 的提交格式，也算是与大厂同步；

第二种方式虽然自由，但是也不比较麻烦，要配置模板。

因此，我们只模仿 `type(scope): subject` 的提交格式，不使用工具 or 模板校验，靠大家自觉遵守即可。

## 格式

`type: description`

## 1. type 类型

type 是 commit 的类别，只允许如下几种标识：

- init: 初始化项目文件夹
- fix: 修复bug
- add: 新功能
- update: 更新
- style : 代码格式改变
- test: 增加测试代码
- revert: 撤销上一次的commit
- build: 构建工具或构建过程等的变动，如：gulp 换成了 webpack，webpack 升级等

## 2. description

description 是对本次提交的简短描述。

不超过50个字符。