一般注意选择type即可：

|  |  |
| --- | --- |
| **feat** | A new feature |
| **fix** | A bug fix |
| **improvement** | An improvement to a current feature |
| **docs** | Documentation only changes |
| **style** | Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc) |
| **refactor** | A code change that neither fixes a bug nor adds a feature |
| **perf** | A code change that improves performance |
| **test** | Adding missing tests or correcting existing tests |
| **build** | Changes that affect the build system or external dependencies (example scopes: gulp, broccoli, npm) |
| **ci** | Changes to our CI configuration files and scripts (example scopes: Travis, Circle, BrowserStack, SauceLabs) |
| **chore** | Other changes that don't modify src or test files |
| **revert** | Reverts a previous commit |

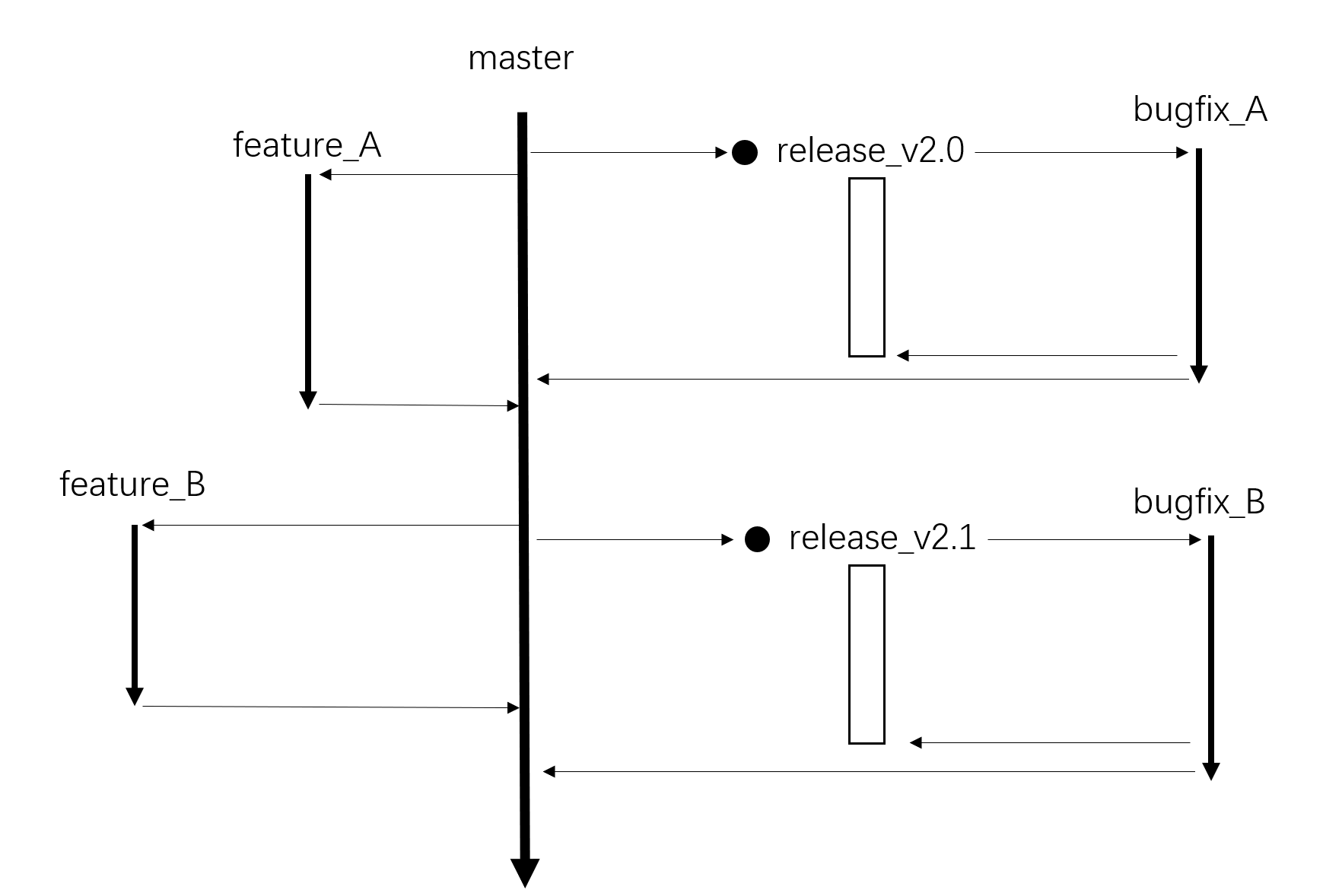
详细规范：

<https://www.conventionalcommits.org/en/v1.0.0/>

eg：

git commit -m “feat(log) 添加日志模块”

使用Git作为代码仓库管理，采用主线发版，分支开发的策略。



feature作为迭代的需求开发分支，开发完成后合回master；release作为发版分支；bugfix\_X作为线上问题修复分支，修复完成后合回release分支和master。