

Commit messages

Commit Message Format

Each commit message consists of a **header**, a **body** and a **footer**. The header has a special format that includes a **type**, a **scope** and a **subject**:

```
<type>(<scope>): <subject>
<BLANK LINE>
<body>
<BLANK LINE>
<footer>
```

The **header** is mandatory and the **scope** of the header is optional.

Any line of the commit message cannot be longer 100 characters! This allows the message to be easier to read on GitHub as well as in various git tools.

Revert

If the commit reverts a previous commit, it should begin with **revert:**, followed by the header of the reverted commit. In the body it should say: **This reverts commit <hash>.**, where the hash is the SHA of the commit being reverted. A commit with this format is automatically created by the `[git revert]``[git-revert]` command.

Type

Must be one of the following:

- **feat**: A new feature
- **fix**: A bug fix
- **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 or correcting existing tests
- **chore**: Changes to the build process or auxiliary tools and libraries such as documentation generation

Scope

The scope could be anything specifying place of the commit change. For example `$location`, `$browser`, `$compile`, `$rootScope`, `ngHref`, `ngClick`, `ngView`, etc...

You can use `*` when the change affects more than a single scope.

Subject

The subject contains succinct description of the change:

- use the imperative, present tense: “change” not “changed” nor “changes”
- don’t capitalize first letter
- no dot (.) at the end

Body

Just as in the **subject**, use the imperative, present tense: “change” not “changed” nor “changes”. The body should include the motivation for the change and contrast this with previous behavior.

Footer

The footer should contain any information about **Breaking Changes** and is also the place to [reference GitHub issues that this commit closes][closing-issues].

Breaking Changes should start with the word **BREAKING CHANGE:** with a space or two newlines. The rest of the commit message is then used for this.

Adapted from:

Tools

- commitlint
-
- (<https://github.com/erlang/otp/wiki/writing-good-commit-messages>)
- <https://git-scm.com/book/en/v2/Distributed-Git-Contributing-to-a-Project>
- <https://gist.github.com/robertpainsi/b632364184e70900af4ab688decf6f53>
- <https://writingfordevelopers.substack.com/p/how-to-write-commit-messages>
- <https://github.com/angular/angular.js/blob/master/DEVELOPERS.md#-git-commit-guideline>
- <https://medium.com/compass-true-north/writing-good-commit-messages-fc33af9d6321>

Commitzen

- Commitzen Cli
- Commitzen VScode extension