

Git Guidelines

Commit

The commit message should be structured as follows:

```
<type>(<scope>): <description>
```

```
[optional <body>]
```

```
[optional <footer>]
```

<type>

Must be one of the following:

- **feat** : A new feature
- **fix** : A bug fix
- **perf** : A code change that improves performance
- **refactor** : A code change that neither fixes a bug nor adds a feature
- **test** : Adding missing tests or correcting existing tests
- **docs** : Documentation only changes
- **chore** : Changes that affect the build system or external dependencies (other than source code changes)

<scope>

The scope should be the name of the object affected. The following is the list of supported scopes:

- **component** : Custom components
- **plugin** : Custom and 3rd party plugins
- **directive** : Custom directives
- **layout** : Page layouts
- **helper** : Utilities and small reusable functions
- **service** : Complex reusable logic
- **api** : Api functions
- **i18n** : Internationalization (languages)

- **style** : UI changes
- **asset** : Fonts, images etc
- **other** : Scopes other than the ones listed above

<description>

The description should be brief and clear about the change:

- should not longer than 70 characters
- use the imperative, present tense: “change” not “changed” nor “changes”
- don’t capitalize the first letter
- no dot (.) at the end

<body>

- A longer commit body MAY be provided one blank line after the short description, providing additional contextual information about the code changes.
- Breaking changes MUST be indicated at the very beginning of the body section. A breaking change MUST consist of the uppercase text BREAKING CHANGE, followed by a colon and a space.

<footer>

- A footer MAY be provided one blank line after the body (or after the description if body is missing).
- The footer MUST only contain external links, issue references, and other meta-information.

Example

- Without body & footer

```
feat(component): add new base component (BaseButton)
```

- with body

```
fix(plugin): change the date format of the $date plugin
```

```
BREAKING CHANGE: change the date format from "06/05/2021" to "06 MAY 2021"
```

- with footer

```
perf(helper): improve array sort speed
```

```
Fixes #24
```

GitHub Guidelines

1. Go to the remote repository <repository_url>
2. Clone the repository (`git clone <repository_url>`)
3. Don't do any modifications to the "main" branch
4. Create a new branch (`git branch <branch_name>`)
5. Switch to the newly created branch (`git switch <branch_name>`)
6. Do what you need to do and commit your changes (`refer commit guidelines`)
7. Switch to the main branch (`git switch main`)
8. Pull the latest changes from the remote branch (`git pull`)
9. Switch back to the newly created branch (`git switch <branch_name>`)
10. Rebase you branch i.e merge all the changes from the main branch to the new branch (`git rebase main`)
11. Resolve all the conflicts, if any
12. Push the new branch to the remote repository (`git push -u origin <branch_name>` or `git push`)
13. Go to the remote repository and open a pull request
14. Repository owner will review your changes and merge it to the main branch
15. Delete the newly created branch after a successful merge (`git branch -d <branch_name>`)