Git commit message Cheat Sheet

by py6jlb via cheatography.com/179964/cs/37439/

Шаблон сообщения

```
!type(?scope): !subject
<?body>
<?footer>
```

type: типы коммитов

test	указывает на любое создание или изменение кода тестов. Пример — создание модульных тестов.
feat	указывает на разработку новой функции для проекта. Примеры: добавление сервиса, функциональности, конечной точки и т.д.
refactor	используется, когда происходит рефакторинг кода, не влияющий на логику/правила системы. Пример — изменения после ревью кода.
style	используется при изменениях форматирования и стиля кода, которые никак не меняют систему. Примеры: смена руководства по стилю или соглашения о линтинге, исправление отступов, удаление пробелов, удаление комментариев и т.д.…
fix	используется при исправлении ошибок, которые порождают баги в системе. Пример — применение обработки для функции, которая ведет себя не так, как ожидалось, и возвращает ошибку.
chore	указывает на изменения в проекте, которые не влияют на систему или тестовые файлы. Это изменения, связанные с разработкой. Примеры: изменение правил для eslint, добавление prettier, добавление расширений файлов в .gitignore.

type: типы коммитов (cont)

docs	используется при изменениях в документации проекта. Пример: добавление сведений в документацию API, измение README и т.д.
build	используется для указания изменений, которые влияют на процесс сборки проекта, или внешних зависимостей. Примеры: Gulp, добавление/удаление зависимостей npm и т.д..
perf	указывает на изменение, которое улучшает производительность системы. Пример — замена ForEach на While.
ci	используется для указания на изменения в конфигурационных файлах CI. Примеры: Circle, Travis, BrowserStack и т.д.
revert	указывает на отмену предыдущего коммита.

subject: сообщение коммита

Мы сообщаем нашей команде, что сделает коммит, если его применить. В английском языке нужно использовать повелительное наклонение, а не прошедшее время.
Пример: «If applied, this commit will…».
Аналог на русском: «При применении этот коммит <что сделает?>».

scope: контекст коммита

Прочитав тип коммита и его сообщение, разработчик уже легко поймет, какое изменение было внесено и что принесет этот коммит, если его применить.
Несмотря на то, что атрибут scope не является обязательным, его можно использовать для добавления контекста. Это позволит сделать сообщение коммита максимально кратким и лаконичным. Помните, что область действия нужно указывать в скобках. Разделяются скоупы символом /.

Прмер :git commit -m "feat(UserService): добавляет /getAppointments эндпоит"

Cheatography

Git commit message Cheat Sheet

by py6jlb via cheatography.com/179964/cs/37439/

Примечания

Для каждого коммита указывается только один тип.

type — обязательный атрибут.

Если вы не знаете, какой тип использовать, вероятно, это большое изменение, и можно разделить этот коммит на два или на большее число коммитов.

Разница между build и chore может быть довольно тонкой, что может привести к путанице. Поэтому важно знать, какой тип когда использовать. В случае с Node.js, например, мы можем считать, что когда происходит добавление/изменение определенной зависимости разработки, присутствующей в devDependencies, мы используем chore. Для изменений/добавлений общих зависимостей проекта, которые оказывают прямое и реальное влияние на систему, мы используем build.

Прмеры

git commit -m "test: добавляет тесты проверки автоматического создания продукта"

git commit -m "feat: добавляет реализацию сервиса отслеживания продуктов"

git commit -m "chore: добавляет правило no-ubdef в eslintrc.json"

By **py6jlb**
cheatography.com/py6jlb/

Published 2nd March, 2023.
Last updated 2nd March, 2023.