

Specification

Спецификация расширяет [Conventional Commits](#). Некоторые правила из базовой спецификации опущены в этом списке для удобства чтения, но также являются актуальными. Слова «MUST», «MUST NOT», «REQUIRED», «SHALL», «MAY» и «OPTIONAL» в данном документе должны интерпретироваться как в RFC 2119.

Коммиты должны (MUST) начинаться с типа, который является существительным: feat, fix и т. д. За ним следует необязательный (OPTIONAL) контекст, необязательный (OPTIONAL) восклицательный знак (!) и обязательные (REQUIRED) двоеточие (:) и пробел ().

Коммит должен (MUST) содержать один из указанных типов.

Контекст может (MAY) следовать после типа. Контекст должен (MUST) быть существительным, заключённым в круглые скобки, описывающий часть кодовой базы, которую затронул коммит. Например, fix(parser).

Описание должно (MUST) следовать сразу за двоеточием (:) и пробелом () после типа или контекста. Описание представляет собой краткое изложение изменений кода. Например, fix: array parsing issue when multiple spaces were contained in string.

Описание должно быть (MUST) не более 72 символов включая тип и контекст (желательно менее 50 символов).

Тело коммита может (MAY) следовать после короткого описания, добавляя дополнительную контекстную информацию об изменениях в коде. Тело должно (MUST) отделяться от описания одной пустой строкой.

Тело коммита имеет произвольную форму и может (MAY) состоять из любого количества абзацев, разделённых новой строкой, каждая строка не должна быть более 72 символов.

Критические изменения должны (MUST) быть указаны в типе, контексте или сноске коммита.

Specification (cont)

Если BREAKING CHANGE включено в сноску, то оно должно (MUST) состоять из прописного текста BREAKING CHANGE, за которым следует двоеточие (:), пробел () и описание. Например, BREAKING CHANGE: environment variables now take precedence over config files.

Если критические изменения находятся в типе или контексте, то они должны (MUST) быть обозначены восклицательным знаком (!), непосредственно перед двоеточием (:). Если используется восклицательный знак (!), то BREAKING CHANGE может (MAY) быть опущен в сноске, а описание коммита должно (SHALL) использоваться для описания критического изменения.

Коммит должен (MUST) быть написан в нижнем регистре, кроме акронимов и слов относящихся к коду (названия классов и т.п.).

В последней строке могут (MAY) содержаться ссылки на задачи в гитлабе, если таких задач несколько, то они должны (MUST) быть перечислены через запятую, например: ref: exampleproject/common#211, exampleproject/web#2.

Types

feat	Features	Добавление нового функционала
fix	Bug Fixes	Исправление багов
style	Stylistic Changes	Стилистическое исправление кода, которое не меняет его поведение (пробелы, точки с запятой, отступы и т.п.)
perf	Performance Improvements	Изменение кода, которое делается для улучшения его производительности



Types (cont)

refactor	Code Refactoring	Изменение кода, которое не вносит новый функционал и не исправляет баги
test	Tests	Добавление новых тестов или исправление уже существующих
ci	Continuous Integrations	Любые изменения связанные с CI/CD
docs	Documentation	Все что касается документации (в том числе внутри кода)
chore	Chores	Любые изменения не модифицирующие исходных код или тесты, например настройка линтеров
vendor	Dependencies Changes	Обновление и изменение зависимостей
revert	Reverts	Возврат на предыдущий коммит
release	Release	Зарезервировано под CI, используется при обновлении версии и генерации changelog файла (не использовать вручную)

Данные типы соответствуют соглашению [Conventional Commits](#) и являются вариацией их [конфига](#).

Commit message structure

Сообщения коммитов должны быть следующей структуры:

```
<type>[(optional scope)]: <description>
```

```
[optional body]
```

```
ref: <gitlab_issue_reference>
```

Commit message examples

Коммит без развернутого описания:

```
chore(ads): add admitad banner
```

```
ref: exampleproject/common#211
```

Коммит с описанием и ссылкой на несколько задач:

```
refactor: change PHP version to 7.4
```

```
refactor to use PHP types not available in PHP 7.2
```

```
ref: exampleproject/common#211, exampleproject/-web#2
```



By **Admitad/Projects**
(admitad)
cheatography.com/admitad/

Published 12th August, 2021.
Last updated 12th August, 2021.
Page 2 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>