

# CONVENTIONAL COMMITS QUICK REFERENCE

Commit message format, types, scopes, breaking changes

## Format

### Commit Message Structure

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

### Parts Explained

<b>type</b>	Category of change (feat, fix, etc.)
<b>scope</b>	Section of codebase affected (optional)
<b>description</b>	Short summary in imperative mood
<b>body</b>	Detailed explanation (optional, after blank line)
<b>footer</b>	Metadata like BREAKING CHANGE or issue refs

### Rules

<b>Imperative mood</b>	"add feature" not "added feature"
<b>Lowercase type</b>	feat: not Feat:
<b>No period</b>	Description does not end with "."
<b>Blank line</b>	Separate body/footer from description

## Types

### Core Types (SemVer-relevant)

<b>feat</b>	New feature (triggers MINOR version bump)
<b>fix</b>	Bug fix (triggers PATCH version bump)

### Extended Types (Common Convention)

<b>build</b>	Build system or external dependencies
<b>chore</b>	Maintenance tasks (no src/test changes)
<b>ci</b>	CI configuration and scripts
<b>docs</b>	Documentation only changes
<b>perf</b>	Performance improvement
<b>refactor</b>	Code change that neither fixes nor adds
<b>revert</b>	Reverts a previous commit
<b>style</b>	Formatting, whitespace (not CSS styling)
<b>test</b>	Adding or correcting tests

## Scope

### Scope Usage

```
feat(auth): add OAuth2 login flow
fix(parser): handle empty input gracefully
docs(readme): update installation steps
refactor(api): extract validation middleware
```

### Scope Guidelines

<b>Module name</b>	feat(auth); fix(parser):
<b>Layer name</b>	feat(api); fix(db):
<b>Feature area</b>	feat(search); fix(checkout):
<b>Dependency</b>	build(deps); chore(npm):
<b>Omit if broad</b>	Use no scope for cross-cutting changes

## Breaking Changes

### Marking Breaking Changes

```
feat!: remove deprecated login endpoint
feat(api)!: change response format to JSON:API
fix!: drop Node 14 support
```

### Footer-Style Breaking Change

```
feat(api): change user endpoint response
BREAKING CHANGE: response now returns array
instead of object. Update client parsing.
```

### Rules

<b>! after type/scope</b>	Shorthand breaking change marker
<b>BREAKING CHANGE:</b>	Footer token (always uppercase)
<b>BREAKING - CHANGE:</b>	Also valid (hyphen form)
<b>SemVer impact</b>	Triggers MAJOR version bump
<b>Any type</b>	Breaking changes can apply to any type

## Examples

### Simple Commits

```
feat: add email notifications
fix: prevent race condition in checkout
docs: correct typo in contributing guide
style: format with prettier
refactor: simplify error handling logic
```

### With Scope

```
feat(blog): add comment threading
fix(auth): refresh token before expiry
test(api): add missing edge case coverage
ci(github): add Node 20 to test matrix
```

### With Body and Footer

```
fix(parser): handle nested quotes correctly
```

Previously, nested quotes caused the parser to enter an infinite loop. This adds a depth counter to prevent unbounded recursion.

Closes #234

## Footer

### Footer Tokens

<b>BREAKING CHANGE:</b>	Describes breaking API change
<b>Closes #123</b>	Auto-close issue on merge
<b>Fixes #456</b>	Auto-close issue (fix reference)
<b>Refs #789</b>	Reference issue without closing
<b>Reviewed-by: name</b>	Reviewer attribution
<b>Co-authored-by: name</b>	Co-author attribution

### Multiple Footers

```
feat(api)!: redesign authentication flow
```

Migrate from session-based to JWT auth.

```
BREAKING CHANGE: /auth endpoints changed
```

Closes #101  
Refs #98, #99

## Tooling

## Commit Linting

```
npm install -D @commitlint/cli \
  @commitlint/config-conventional
echo "module.exports = { extends: \
  ['@commitlint/config-conventional'] }" \
  > commitlint.config.js
```

## Popular Tools

<b>commitlint</b>	Lint commit messages against convention
<b>husky</b>	Git hooks (run commitlint on commit)
<b>commitizen (cz)</b>	Interactive commit message builder
<b>standard-version</b>	Auto changelog + version bump
<b>semantic-release</b>	Fully automated release pipeline
<b>release-please</b>	Google's release automation tool

## Commitizen Setup

```
npm install -D commitizen \
  cz-conventional-changelog
npx commitizen init cz-conventional-changelog
# Use: npx cz (or git cz with alias)
```

## Common Patterns

### Version Bump Mapping

<b>fix:</b>	PATCH (1.0.0 → 1.0.1)
<b>feat:</b>	MINOR (1.0.0 → 1.1.0)
<b>BREAKING CHANGE</b>	MAJOR (1.0.0 → 2.0.0)
<b>docs:, style:, etc.</b>	No version bump

### Changelog Grouping

```
## [1.2.0] - 2026-03-27
### Features
- add email notifications (abc1234)
### Bug Fixes
- prevent race condition (#123) (def5678)
```

### Revert Format

```
revert: feat(blog): add comment threading
This reverts commit abc1234def5678.
```