

# Conventional Commits 快速参考

提交格式、类型、作用域、破坏性变更

## 格式

### 提交消息结构

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

### 各部分说明

<b>type</b>	变更类别 (feat、fix 等)
<b>scope</b>	受影响的代码区域 (可选)
<b>description</b>	用祈使语气写的简短摘要
<b>body</b>	详细说明 (可选, 空行后)
<b>footer</b>	元数据, 如 BREAKING CHANGE 或 issue 引用

### 规则

<b>Imperative mood</b>	用 "add feature" 而非 "added feature"
<b>Lowercase type</b>	feat: 而非 Feat:
<b>No period</b>	描述末尾不加句号
<b>Blank line</b>	body/footer 与描述之间空一行

## 类型

### 核心类型 (影响 SemVer)

<b>feat</b>	新功能 (触发 MINOR 版本升级)
<b>fix</b>	Bug 修复 (触发 PATCH 版本升级)

### 扩展类型 (通用约定)

<b>build</b>	构建系统或外部依赖
<b>chore</b>	维护任务 (不涉及源码/测试变更)
<b>ci</b>	CI 配置与脚本
<b>docs</b>	仅文档变更
<b>perf</b>	性能优化
<b>refactor</b>	既不修复 bug 也不添加功能的代码重构
<b>revert</b>	回退某次提交
<b>style</b>	格式、空白 (非 CSS 样式)
<b>test</b>	添加或修正测试

## 作用域

### 作用域用法

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

### 作用域命名指南

<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>	跨模块变更时省略作用域

## 破坏性变更

### 标记破坏性变更

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

### Footer 形式的破坏性变更

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

## 规则

<b>! after type/scope</b>	破坏性变更的简写标记
<b>BREAKING CHANGE:</b>	Footer token (必须全大写)
<b>BREAKING-CHANGE:</b>	同样有效 (连字符形式)
<b>SemVer impact</b>	触发 MAJOR 版本升级
<b>Any type</b>	任何类型都可以有破坏性变更

## 示例

### 简单提交

```
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
```

### 带作用域

```
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
```

### 带 body 和 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 Token

<b>BREAKING CHANGE:</b>	描述破坏性 API 变更
<b>Closes #123</b>	合并时自动关闭 issue
<b>Fixes #456</b>	自动关闭 issue (fix 引用)
<b>Refs #789</b>	引用 issue 但不关闭
<b>Reviewed-by: name</b>	审阅者署名
<b>Co-authored-by: name</b>	共同作者署名

### 多个 Footer

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

Migrate from session-based to JWT auth.

BREAKING CHANGE: /auth endpoints changed
Closes #101
Refs #98, #99
```

## 工具链

### 提交 Lint

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

### 常用工具

<b>commitlint</b>	按规范 lint 提交消息
<b>husky</b>	Git hooks (提交时运行 commitlint)
<b>commitizen (cz)</b>	交互式提交消息生成器
<b>standard-version</b>	自动生成 changelog + 版本升级
<b>semantic-release</b>	全自动发布流水线
<b>release-please</b>	Google 的发布自动化工具

## Commitizen 设置

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

## 常用模式

### 版本升级映射

<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: 等</b>	不升级版本

### Changelog 分组

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

### 回退格式

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