

/sentry-commit

Source: `~/ .claude/skills/sentry-commit/SKILL.md`

name: commit **description:** Create commit messages following Sentry conventions. Use when committing code changes, writing commit messages, or formatting git history. Follows conventional commits with Sentry-specific issue references.

Sentry Commit Messages

Follow these conventions when creating commits for Sentry projects.

Prerequisites

Before committing, ensure you're working on a feature branch, not the main branch.

```
# Check current branch
git branch --show-current
```

If you're on `main` or `master`, create a new branch first:

```
# Create and switch to a new branch
git checkout -b <type>/<short-description>
```

Branch naming should follow the pattern: `<type>/<short-description>` where type matches the commit type (e.g., `feat/add-user-auth`, `fix/null-pointer-error`, `ref/extract-validation`).

Format

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

<body>

<footer>
```

The header is required. Scope is optional. All lines must stay under 100 characters.

Commit Types

Type	Purpose
<code>feat</code>	New feature
<code>fix</code>	Bug fix
<code>ref</code>	Refactoring (no behavior change)
<code>perf</code>	Performance improvement
<code>docs</code>	Documentation only
<code>test</code>	Test additions or corrections
<code>build</code>	Build system or dependencies
<code>ci</code>	CI configuration
<code>chore</code>	Maintenance tasks
<code>style</code>	Code formatting (no logic change)
<code>meta</code>	Repository metadata
<code>license</code>	License changes

Subject Line Rules

- Use imperative, present tense: "Add feature" not "Added feature"
- Capitalize the first letter
- No period at the end
- Maximum 70 characters

Body Guidelines

- Explain **what** and **why**, not how
- Use imperative mood and present tense
- Include motivation for the change
- Contrast with previous behavior when relevant

Footer: Issue References

Reference issues in the footer using these patterns:

```
Fixes GH-1234
Fixes #1234
Fixes SENTRY-1234
Refs LINEAR-ABC-123
```

- `Fixes` closes the issue when merged
- `Refs` links without closing

AI-Generated Changes

When changes were primarily generated by a coding agent (like Claude Code), include the Co-Authored-By attribution in the commit footer:

```
Co-Authored-By: Claude <noreply@anthropic.com>
```

This is the only indicator of AI involvement that should appear in commits. Do not add phrases like "Generated by AI", "Written with Claude", or similar markers in the subject, body, or anywhere else in the commit message.

Examples

Simple fix

fix(api): Handle null response in user endpoint

The user API could return null for deleted accounts, causing a crash in the dashboard. Add null check before accessing user properties.

Fixes SENTRY-5678

Co-Authored-By: Claude <noreply@anthropic.com>

Feature with scope

feat(alerts): Add Slack thread replies for alert updates

When an alert is updated or resolved, post a reply to the original Slack thread instead of creating a new message. This keeps related notifications grouped together.

Refs GH-1234

Refactor

ref: Extract common validation logic to shared module

Move duplicate validation code from three endpoints into a shared validator class. No behavior change.

Breaking change

feat(api)!: Remove deprecated v1 endpoints

Remove all v1 API endpoints that were deprecated in version 23.1. Clients should migrate to v2 endpoints.

BREAKING CHANGE: v1 endpoints no longer available

Fixes SENTRY-9999

Revert Format

```
revert: feat(api): Add new endpoint
```

This reverts commit abc123def456.

Reason: Caused performance regression in production.

Principles

- Each commit should be a single, stable change
- Commits should be independently reviewable
- The repository should be in a working state after each commit

References

- [Sentry Commit Messages](#)

Revision #4

Created 2026-02-18 08:39:59 UTC by John

Updated 2026-05-31 20:01:30 UTC by John