/sentry-create-pr

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

---

name: create-pr description: Create pull requests following Sentry conventions. Use when opening PRs, writing PR descriptions, or preparing changes for review. Follows Sentry's code review guidelines.

Create Pull Request

Create pull requests following Sentry's engineering practices.

Requires: GitHub CLI (`gh`) authenticated and available.

Prerequisites

Before creating a PR, ensure all changes are committed. If there are uncommitted changes, run the `sentry-skills:commit` skill first to commit them properly.

```
# Check for uncommitted changes
git status --porcelain
```

If the output shows any uncommitted changes (modified, added, or untracked files that should be included), invoke the `sentry-skills:commit` skill before proceeding.

Process

Step 1: Verify Branch State

```
# Detect the default branch
BASE=$(gh repo view --json defaultBranchRef --jq '.defaultBranchRef.name')

# Check current branch and status
git status
git log $BASE..HEAD --oneline
```

Ensure:

- All changes are committed
- Branch is up to date with remote
- Changes are rebased on the base branch if needed

Step 2: Analyze Changes

Review what will be included in the PR:

```
# See all commits that will be in the PR
git log $BASE..HEAD

# See the full diff
git diff $BASE...HEAD
```

Understand the scope and purpose of all changes before writing the description.

Step 3: Write the PR Description

Use this structure for PR descriptions (ignoring any repository PR templates):

```
<brief description of what the PR does>

<why these changes are being made - the motivation>
```

```
<alternative approaches considered, if any>


<any additional context reviewers need>
```

Do NOT include:

- "Test plan" sections
- Checkbox lists of testing steps
- Redundant summaries of the diff

Do include:

- Clear explanation of what and why
- Links to relevant issues or tickets
- Context that isn't obvious from the code
- Notes on specific areas that need careful review

Step 4: Create the PR

```
gh pr create --draft --title "<type>(<scope>): <description>" --body "$(cat <<'EOF'
<description body here>
EOF
)"
```

Title format follows commit conventions:

- `feat(scope): Add new feature`
- `fix(scope): Fix the bug`
- `ref: Refactor something`

PR Description Examples

Feature PR

```
Add Slack thread replies for alert notifications

When an alert is updated or resolved, we now post a reply to the original
Slack thread instead of creating a new message. This keeps related
notifications grouped and reduces channel noise.
```

```
Previously considered posting edits to the original message, but threading
better preserves the timeline of events and works when the original message
is older than Slack's edit window.

Refs SENTRY-1234
```

Bug Fix PR

```
Handle null response in user API endpoint

The user endpoint could return null for soft-deleted accounts, causing
dashboard crashes when accessing user properties. This adds a null check
and returns a proper 404 response.

Found while investigating SENTRY-5678.

Fixes SENTRY-5678
```

Refactor PR

```
Extract validation logic to shared module

Moves duplicate validation code from the alerts, issues, and projects
endpoints into a shared validator class. No behavior change.

This prepares for adding new validation rules in SENTRY-9999 without
duplicating logic across endpoints.
```

Issue References

Reference issues in the PR body:

Syntax	Effect
`Fixes #1234`	Closes GitHub issue on merge
`Fixes SENTRY-1234`	Closes Sentry issue

Syntax	Effect
`Refs GH-1234`	Links without closing
`Refs LINEAR-ABC-123`	Links Linear issue

Guidelines

- **One PR per feature/fix** - Don't bundle unrelated changes
- **Keep PRs reviewable** - Smaller PRs get faster, better reviews
- **Explain the why** - Code shows what; description explains why
- **Mark WIP early** - Use draft PRs for early feedback

Editing Existing PRs

If you need to update a PR after creation, use `gh api` instead of `gh pr edit`:

```
# Update PR description
gh api -X PATCH repos/{owner}/{repo}/pulls/PR_NUMBER -f body="$(cat <<'EOF'
Updated description here
EOF
)"

# Update PR title
gh api -X PATCH repos/{owner}/{repo}/pulls/PR_NUMBER -f title='new: Title here'

# Update both
gh api -X PATCH repos/{owner}/{repo}/pulls/PR_NUMBER \
  -f title='new: Title' \
  -f body='New description'
```

Note: `gh pr edit` is currently broken due to GitHub's Projects (classic) deprecation.

References

- Sentry Code Review Guidelines
- Sentry Commit Messages

Revision #4
Created 2026-02-18 08:39:59 UTC by John

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