



FM

FLOWMATRIX.DEV

Premium Workflow

A practical guide – step by step

Release Notes Automation

Turn merged pull requests into clear release notes and publish them to Slack and Notion.

DEVELOPER TOOLS

INTERMEDIATE

ABOUT 2 HOURS SETUP

OVERVIEW

Release Notes Automation

Turn merged pull requests into clear release notes and publish them to Slack and Notion.

CATEGORY

Developer Tools

DIFFICULTY

Intermediate

DURATION

About 2 hours setup

What you use it for

Consistent release notes without manual changelog work.

Tools in this workflow

GitHub Alternatives · OpenAI API · Slack · Notion API

How to implement it

1

Collect merged pull requests

Fetch merged pull requests for the release range from the GitHub API.

WHAT YOU NEED TO DO

1. Create a read-only GitHub token.
2. List pull requests merged since the last tag.
3. Ignore dependency and bot pull requests.
4. Store the raw list for the next step.

CHECK AT THE END

- [] Token has read access
- [] Bot pull requests are filtered

RESULT

A clean list of merged pull requests is ready.

LINKS

GitHub REST API: <https://docs.github.com/en/rest>

2

Group changes by type

Sort pull requests into features, fixes and internal changes before writing.

WHAT YOU NEED TO DO

1. Read labels and titles from each pull request.
2. Map labels to feature, fix or internal.
3. Flag pull requests without a clear label.
4. Save the grouped list for review.

CHECK AT THE END

- [] Every pull request has a group
- [] Unclear labels are flagged

RESULT

Changes are grouped in a predictable structure.

LINKS

GitHub Issues API: <https://docs.github.com/en/rest/issues>

3

Draft release notes

Ask the model to write short, user-facing notes from the grouped changes.

WHAT YOU NEED TO DO

1. Send grouped changes to the OpenAI API.
2. Ask for one sentence per user-facing change.
3. Exclude internal changes from the public draft.
4. Reject drafts with unknown product names.

CHECK AT THE END

- [] Draft uses plain language

[] Internal changes are excluded

RESULT

A readable release note draft exists.

LINKS

OpenAI Text Generation: <https://platform.openai.com/docs/guides/text-generation>

4

Review and approve

Let a maintainer review the draft before anything is published.

WHAT YOU NEED TO DO

1. Post the draft in a Slack review channel.
2. Ask for approval with a short checklist.
3. Apply requested edits in Notion.
4. Mark the draft as approved.

CHECK AT THE END

- ☐ A maintainer approved the draft
- ☐ Edits are reflected in the final text

RESULT

The release notes are approved and final.

LINKS

Slack chat.postMessage: <https://api.slack.com/methods/chat.postMessage>

5

Publish and archive

Send the final notes to Slack and store a copy in Notion for future reference.

WHAT YOU NEED TO DO

1. Post the final notes to the release channel.
2. Create a Notion page for the release.
3. Link the GitHub tag in the page.
4. Archive the draft after publishing.

CHECK AT THE END

- ☐ Slack post links to the release
- ☐ Notion page is stored

RESULT

The release is published and archived.

LINKS

Notion API Reference: <https://developers.notion.com/reference>

Recommended Tools & Resources

All tools in this workflow with their official website.

GitHub Alternatives

<https://github.com/>

OpenAI API

<https://platform.openai.com/?ref=flowmatrix>

Official URL

Slack

<https://slack.com/>

Notion API

<https://www.notion.so/?ref=flowmatrix>

Official URL

Full version as Premium PDF

All steps, tools and resources as a PDF – €4.99.

[Buy now – €4.99](#)