

Collaborating with Your Community on GitHub

Templates, Forms, and Practical Workflows

How to use this guide. Drop the shown files into your repository (or a central `.github` repository). All examples are ASCII-only and ready to copy. When building this PDF, compile with `-shell-escape` so the `minted` code blocks render.

Contents

1	Purpose	2
2	Ready-to-use Repository Layout	2
3	Issue <i>Form</i> (YAML)	3
3.1	Bug Report Form	3
4	Issue <i>Template</i> (Markdown)	4
4.1	Feature Request Template (Markdown)	4
5	Issue Template Config	4
6	Pull Request Template	5
6.1	Single Template (Markdown)	5
6.2	Multiple PR Templates	5
7	Discussion Category Form	6
8	Centralization & Overrides	6
9	Troubleshooting & Gotchas	6
10	Quick Start Checklists	7
11	Appendix: Copy-ready Snippets	7

1 Purpose

A lightweight, battle-tested starter to make your repository friendly for contributors: issue forms and templates, pull request templates, discussion forms, centralization patterns, and common pitfalls. Everything here assumes the files live on the *default branch* so GitHub can pick them up.

2 Ready-to-use Repository Layout

Tip. Directory and some file names are case-sensitive (e.g., `ISSUE_TEMPLATE`, `discussion_template`).

```
.github/  
  ISSUE_TEMPLATE/  
    bug_report.yml  
    feature_request.md  
    config.yml  
  PULL_REQUEST_TEMPLATE.md  
  discussion_template/  
    ideas.yml
```

3 Issue *Form* (YAML)

Place YAML-based forms in `.github/ISSUE_TEMPLATE/`. Forms render as a guided UI.

Heads-up. YAML indentation matters. Use spaces (no tabs). Keep quotes simple ASCII.

3.1 Bug Report Form

File: `.github/ISSUE_TEMPLATE/bug_report.yml`

```
name: "Bug report"
description: "Report a reproducible problem"
title: "bug: <concise summary>"
labels: ["bug"]
assignees: []
body:
  - type: markdown
    attributes:
      value: |
        Thanks for filing a bug! Please fill out all required fields.
  - type: input
    id: version
    attributes:
      label: Affected version
      description: "Choose from releases if possible."
      placeholder: "v1.2.3"
    validations:
      required: true
  - type: textarea
    id: repro
    attributes:
      label: Steps to reproduce
      description: "What did you do? What happened? What did you expect?"
      placeholder: |
        1) ...
        2) ...
        3) ...
    validations:
      required: true
  - type: textarea
    id: logs
    attributes:
      label: Logs / stacktrace
      description: "Paste relevant logs. Use a gist for long traces."
      render: shell
```

4 Issue *Template* (Markdown)

Markdown templates live alongside forms and appear under the “New issue” flow.

4.1 Feature Request Template (Markdown)

File: `.github/ISSUE_TEMPLATE/feature_request.md`

```
---
name: "Feature request"
about: "Propose an enhancement"
title: "feat: <short title>"
labels: ["enhancement"]
assignees: []
---

## Problem
<!-- What problem are you trying to solve? -->

## Proposal
<!-- What would you like to see happen? -->

## Alternatives
<!-- What did you consider and why is this better? -->

## Additional context
<!-- Links, screenshots, references -->
```

5 Issue Template Config

File: `.github/ISSUE_TEMPLATE/config.yml`

Use cases. Disable “blank issues,” and add dedicated buttons that route users to Discussions, a support portal, or private security reporting.

```
blank_issues_enabled: false
contact_links:
  - name: Ask a question (Discussions)
    url: https://github.com/<owner>/<repo>/discussions
    about: "Use Discussions for Q&A and ideas."
  - name: Security bug bounty
    url: https://security.example.com/bounty
    about: "Report security issues privately."
```

6 Pull Request Template

You may keep a single template or multiple templates via a folder.

6.1 Single Template (Markdown)

File: `.github/PULL_REQUEST_TEMPLATE.md`

```
<!--
Briefly explain the change and link issues (e.g., Closes #123).
The comments in this header will not appear after submission.
-->
```

```
## Summary
```

```
- ...
```

```
## Checklist
```

- [] Tests added/updated
- [] Docs updated
- [] Linked issue(s)

```
## Screenshots / Logs
```

6.2 Multiple PR Templates

Create `.github/PULL_REQUEST_TEMPLATE/` and add several files (for example, `bugfix.md`, `feature.md`).
Share links like:

`https://github.com/<owner>/<repo>/compare/main...my-branch?expand=1&template=feature.md`

7 Discussion Category Form

Discussion forms help funnel ideas and questions with required fields.

Folder: `.github/discussion_template/` **File:** `ideas.yml`

```
title: "Idea: <concise summary>"
labels: ["discussion", "idea"]
body:
  - type: markdown
    attributes:
      value: |
        Use this form to propose an idea. Required fields help us triage quickly.
  - type: textarea
    id: motivation
    attributes:
      label: Motivation
      description: "What problem would this idea solve?"
    validations:
      required: true
  - type: textarea
    id: details
    attributes:
      label: Details
      description: "Sketch the solution, trade-offs, and impact."
```

8 Centralization & Overrides

- **Organization Defaults.** Create an org- or user-level repository named `.github` and place the same folders/files there to apply defaults across all repositories.
- **Per-Repo Overrides.** A repository overrides the org defaults by adding its own files of the same path/name on its default branch.

9 Troubleshooting & Gotchas

1. **Default Branch Only.** Templates/config must live on the repository's default branch.
2. **Case Sensitivity.** `ISSUE_TEMPLATE` and `discussion_template` paths are case-sensitive.
3. **YAML Indentation.** Two spaces per level; no tabs. Validate YAML before committing.
4. **Non-ASCII Characters.** Avoid emojis or special symbols in YAML or Markdown front-matter to prevent tooling issues.
5. **Private Security Reporting.** If private vulnerability reporting is enabled, GitHub surfaces a security disclosure button.¹
6. **Compiling this PDF.** Ensure `pdflatex -shell-escape` (or `xelatex -shell-escape`) so `minted` can call Pygments. If you cannot enable shell-escape, replace `minted` with `verbatim`.

¹Availability depends on your GitHub plan/instance.

10 Quick Start Checklists

For a New Repository

- Create `.github/ISSUE_TEMPLATE/` with one YAML form and one Markdown template.
- Add `config.yml` to disable blank issues and add contact links.
- Add `PULL_REQUEST_TEMPLATE.md`.
- (Optional) Add `discussion_template/` for ideas or Q&A.
- Push to the *default branch*, then verify via the “New issue” & “New pull request” flows.

Hardening the Triage Flow

- Require required fields in forms (e.g., reproduction steps, version).
- Pre-apply labels/assignees to route bugs and features automatically.
- Add contact links for Discussions and private security intake.

11 Appendix: Copy-ready Snippets

Minimal Feature Request (Markdown)

```
---
name: "Feature request"
about: "Propose an enhancement"
title: "feat: <short title>"
labels: ["enhancement"]
---
```

```
## Problem
## Proposal
## Alternatives
## Additional context
```

Minimal PR Template

```
## Summary
- ...

## Checklist
- [ ] Tests
- [ ] Docs
- [ ] Linked issues
```

Minimal Issue Form (YAML)

```
name: "Bug report"
title: "bug: <summary>"
labels: ["bug"]
body:
  - type: input
    id: version
    attributes: { label: Affected version, placeholder: "v1.2.3" }
    validations: { required: true }
  - type: textarea
    id: repro
    attributes: { label: Steps to reproduce }
    validations: { required: true }
```

This document is designed to be copied into your workflow repos and adapted to your project's voice.