

Spickzettel: Pull Requests (PRs) in der GitHub GUI

1. PR erstellen in der GitHub GUI

1. **Zum Repository navigieren**
 - o Gehe auf **GitHub** und öffne dein Repository.
 - o Klicke auf den Tab **Pull Requests**.
 - o Klicke auf **New Pull Request**.
2. **Branches auswählen**
 - o Wähle den **Quellbranch** (z. B. feature-branch).
 - o Wähle den **Zielbranch** (z. B. main oder develop).
 - o GitHub zeigt Unterschiede zwischen den Branches an.
3. **Details hinzufügen**
 - o **Titel:** Kurze Beschreibung der Änderungen.
 - o **Beschreibung:** Detaillierte Infos zu den Änderungen, ggf. Referenzen zu Issues (#123).
 - o **Reviewer zuweisen** (optional): Personen, die den Code prüfen sollen.
 - o **Labels, Milestones und Assignees** setzen (optional).
4. **PR absenden**
 - o Klicke auf **Create Pull Request**.

2. PR-Überprüfung & Diskussion

- **Code-Änderungen einsehen:** Tab **Files changed** zeigt die Unterschiede.
- **Kommentare hinterlassen:**
 - o Zeilenweise im Code oder als allgemeiner Kommentar.
 - o Erwähne Teammitglieder mit @username, um sie zu benachrichtigen.
- **Reviewer geben Feedback:**
 - o Änderungen anfordern (Request changes).
 - o PR genehmigen (Approve).

3. Änderungen aktualisieren

Falls Änderungen nötig sind:

1. Lokale Änderungen durchführen und committen:

```
git add .  
git commit -m "Feedback eingearbeitet"  
git push origin feature-branch
```

2. Die Änderungen erscheinen automatisch im PR.

4. PR mergen

1. Nach Freigabe auf „Merge Pull Request“ klicken.

2. Merge-Strategie wählen:
 - o **Merge Commit:** Standard-Merge mit separatem Merge-Commit.
 - o **Squash & Merge:** Alle Commits zu einem zusammenfassen.
 - o **Rebase & Merge:** Änderungen auf main ohne Merge-Commit übernehmen.
3. **Branch löschen** (empfohlen für Übersichtlichkeit).

5. PR-Branch nach Merge löschen

```
git branch -d feature-branch  
git push origin --delete feature-branch
```

6. Konflikte in PRs lösen

Falls GitHub Merge-Konflikte erkennt:

1. Auf „Resolve conflicts“ klicken.
2. **Dateien manuell bearbeiten** (Konfliktmarkierungen <<<<<<, =====, >>>>>> entfernen).
3. **Änderungen speichern & als gelöst markieren.**
4. **PR erneut zur Überprüfung senden.**

Best Practices

- **PRs klein & thematisch halten**, um Reviews zu erleichtern.
- **Aussagekräftige Titel & Beschreibungen** schreiben.
- **Automatische Tests in GitHub Actions einbinden**, um Fehler frühzeitig zu erkennen.
- **Branches nach dem Merge löschen**, um das Repository sauber zu halten.