

Spickzettel: Git Stash – Änderungen zwischenspeichern

1. Was ist `git stash`?

- Speichert uncommittete Änderungen temporär, ohne sie zu committen.
- Ermöglicht einen sauberen Wechsel zu anderen Branches oder Aufgaben.
- Änderungen können später wiederhergestellt oder verworfen werden.

2. Änderungen zwischenspeichern

`git stash`

- Speichert alle uncommitteten Änderungen und setzt das Arbeitsverzeichnis zurück.

`git stash push -m "Meine gespeicherten Änderungen"`

- Speichert die Änderungen mit einer Beschreibung (empfohlen für bessere Übersicht).

3. Gespeicherte Stashes anzeigen

`git stash list`

- Zeigt eine Liste aller zwischengespeicherten Änderungen.

4. Änderungen aus dem Stash wiederherstellen

`git stash apply`

- Wendet den zuletzt gespeicherten Stash an (bleibt erhalten).

`git stash pop`

- Wendet den zuletzt gespeicherten Stash an und entfernt ihn aus der Liste.

`git stash apply stash@{1}`

- Wendet eine spezifische Stash-Nummer an (`stash@{1}`, `stash@{2}`, ...).

5. Spezifische Datei aus dem Stash wiederherstellen

`git checkout stash@{0} -- <datei>`

- Nur eine bestimmte Datei aus dem Stash wiederherstellen.

6. Stash-Eintrag löschen

```
git stash drop stash@{0}
```

- Löscht einen spezifischen Stash-Eintrag.

```
git stash clear
```

- Entfernt **alle** gespeicherten Stashes (Vorsicht!).

Best Practices

- Vor `git stash`, immer `git status` prüfen, um unerwartete Ergebnisse zu vermeiden.
- `git stash list` regelmäßig verwenden, um alte Stashes zu verwalten.
- Statt einfach `git stash`, besser `git stash push -m "Beschreibung"` für mehr Übersicht.
- Wenn ein Stash nicht mehr benötigt wird, mit `git stash drop` aufräumen.