

Git

ITS-Net-Lin

Sebastian Meisel

7. Januar 2025

1 Git

Git ist ein verteiltes Versionskontrollsystem, das Entwicklern hilft, Änderungen an Dateien zu verfolgen, mit anderen zusammenzuarbeiten und die Historie von Projekten zu verwalten. Es wurde wie der Linux-Kernel von Linus Torvalds entwickelt. Git kann vollständig über die Kommandozeile genutzt werden und bietet viele mächtige Funktionen, um die Arbeit mit Quellcode und Dateien zu erleichtern.

1.1 Grundlegende Git-Befehle

Die wichtigsten Git-Befehle lassen sich in zwei Kategorien einteilen: Befehle für die Arbeit mit einem Repository und Befehle für die Verwaltung von Änderungen.

1.1.1 Git-Repository initialisieren

Bevor Sie Git verwenden können, müssen Sie ein Git-Repository initialisieren. Dies geschieht mit dem folgenden Befehl:

```
1 git init
```

Dieser Befehl erstellt ein neues Git-Repository im aktuellen Verzeichnis. Sie können nun Dateien zu diesem Repository hinzufügen und es mit der Versionskontrolle beginnen.

1.1.2 Dateien zum Repository hinzufügen

Nachdem Sie ein Repository initialisiert haben, können Sie Dateien zum Staging-Bereich hinzufügen, um sie in die Versionskontrolle zu übernehmen:

```
1 git add <dateiname>
```

Um alle Dateien hinzuzufügen, verwenden Sie:

```
1 git add .
```

1.1.3 Änderungen committen

Nachdem die Dateien zum Staging-Bereich hinzugefügt wurden, können Sie diese Änderungen mit einem Commit speichern:

```
1 git commit -m "Beschreibung der Änderung"
```

Der Parameter `-m` ermöglicht es, eine kurze Beschreibung des Commits hinzuzufügen, um die vorgenommenen Änderungen zu dokumentieren.

1.1.4 Git-Status anzeigen

Um den Status des Repositories zu überprüfen und zu sehen, welche Dateien geändert wurden, verwenden Sie:

```
1 git status
```

Dieser Befehl zeigt Ihnen die aktuellen Änderungen und den Staging-Bereich an.

1.1.5 Änderungen an Git-Repository zurücksetzen

Falls Sie Änderungen rückgängig machen möchten, können Sie sie mit folgendem Befehl zurücksetzen:

```
1 git checkout -- <dateiname>
```

Dieser Befehl stellt die Datei wieder her, wie sie im letzten Commit war.

1.2 Arbeiten mit Remote-Repositories

Neben lokalen Git-Repositories können Sie auch mit Remote-Repositories arbeiten, die sich auf einem anderen Server befinden (z. B. GitHub, GitLab).

1.2.1 Remote-Repository hinzufügen

Um ein Remote-Repository zu Ihrem lokalen Repository hinzuzufügen, verwenden Sie den Befehl:

```
1 git remote add origin <url-des-repositories>
```

Beispiel:

```
1 git remote add origin https://github.com/Benutzername/Projekt.git
```

1.2.2 Änderungen an Remote-Repository senden

Nach dem Commit können Sie Ihre Änderungen an das Remote-Repository senden:

```
1 git push origin master
```

1.2.3 Änderungen aus Remote-Repository herunterladen

Um Änderungen aus dem Remote-Repository zu holen, verwenden Sie den Befehl:

```
1 git pull origin master
```