

LU06b - Git

Git ist eine Open-Source-Software zur verteilten Versionskontrolle, die 2005 von Linus Torvalds für den Linux-Kernel entwickelt wurde. Sie ermöglicht es einem Team von Entwicklern, an einem Projekt zu arbeiten, bei dem jeder von ihnen eine Kopie des Haupt-Repositorys besitzt, das sich auf dem zentralen Server befindet.

Merkmale von Git

- **Kostenlos und Open-Source** Sie müssen für die Verwendung von Git nichts bezahlen. Außerdem ist der Quellcode offen zugänglich, so dass Sie ihn nach Ihren Wünschen verändern können.
- **Verteilte Entwicklung** Git unterstützt das Klonen, wobei die Mitwirkenden eine lokale Kopie des Hauptrepositorys speichern können.
- **Unterstützt nicht-lineare Entwicklung** Git unterstützt unabhängige Codezeilen, die als Branches bekannt sind und unabhängig von der Hauptcodebasis bereitgestellt, übertragen und aktualisiert werden können. Die Branches bieten einem Entwickler also einen sicheren Raum, um etwas Neues zu implementieren und zu testen, ohne in die Codebasis einzudringen. Die Branches können später mit der Codebasis zusammengeführt werden.
- **Skalierbar** Git ist gut skalierbar, so dass sich die Anzahl der Mitarbeiter problemlos erhöhen lässt.
- **Sicher** Dieses Tool verwendet die sichere Hash-Funktion (SHA1) zur Benennung und Identifizierung von Objekten in einem Repository. Daher wird jede Änderung gut überwacht, was es unmöglich macht, unbemerkt Änderungen vorzunehmen.

Was ist GitHub?

GitHub ist ein 2008 gegründetes Hosting-Unternehmen, das Entwicklern eine Plattform bietet, auf der sie ihre Projekte hosten und gemeinsam nutzen können. Es bietet Zugriffskontrolle und Verwaltungsfunktionen für jedes gehostete Projekt. Im Grunde genommen bietet es neben anderen Funktionen von Git auch die Verwaltung von Quellcode. Es gibt viele Alternativen zu GitHub wie zum Beispiel GitLab und Bitbucket. Sie brauchen also nicht unbedingt GitHub, um Git zu nutzen, aber Sie brauchen Git, um GitHub zu nutzen.

Git installieren

Sollten Sie den Windows Paketmanager chocolatey installiert haben, können Sie für die Installation von git nun folgenden Befehl auf einer Admin-Konsole ausführen: `choco install git`

Haben Sie chocolatey nicht mehr installiert, so laden Sie sich die neueste Version von Git für Windows [hier](#) herunter und führen Sie den installer aus.

Git Workflow



Nachdem der Benutzer Änderungen an der Arbeitsdatei vorgenommen hat, merkt sich Git die kürzlich geänderten Dateien. Der Benutzer kann die geänderten Dateien mit dem Befehl `git status` überprüfen.

Die geänderten Dateien werden mit `git add` zu einem Staging-Bereich hinzugefügt, der als temporärer Speicherort für die Dateien dient.

Alle Staging-Dateien sind bereit, mit `git commit` in ein lokales Repository übertragen/verschoben zu werden.

Bis zu diesem Zeitpunkt sind die Änderungen nur auf dem lokalen Rechner des Entwicklers sichtbar und können von anderen Mitwirkenden nicht online eingesehen werden. Um die Datei online verfügbar zu machen, müssen Sie sie bei einem Online-Hosting-Dienst wie GitHub hosten. Sie müssen also das lokale Repository mittels `git push` hochladen, um es mit dem entfernten Repository zu synchronisieren.

Nehmen wir an, die anderen Mitwirkenden haben Änderungen auch an das entfernte Repository übertragen. Sie müssen Ihr lokales Projektarchiv mit dem entfernten Projektarchiv synchronisieren, um mit den vorgenommenen Änderungen auf dem neuesten Stand zu sein. Dazu verwenden Sie den Befehl `git pull`, der Ihr lokales Repository synchronisiert.

git status

Der Befehl `git status` listet alle Dateien auf, die in letzter Zeit geändert wurden und noch nicht zum lokalen Repository hinzugefügt worden sind. Im folgenden Fall wurde die Datei `Two Sum.md` noch nicht zum lokalen Repository hinzugefügt.

```
PS C:\Users\Maurizi\Desktop\coding problems>
git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what
  will be committed)
    leetcode/Two Sum.md

  nothing added to commit but untracked files
  present (use "git add" to track)
PS C:\Users\Maurizi\Desktop\coding problems>
```

git add

Der Befehl `git add` wird verwendet, um die Dateien in den Staging-Bereich zu verschieben, damit sie übertragen werden können. Verwenden Sie zum Bereitstellen Ihrer Dateien die folgenden Befehle:

- `git add filename` - zum Hinzufügen einer bestimmten Datei
- `git add -all`, abgekürzt `git add -A`, verschiebt alle Dateien, d. h. die neuen, geänderten und gelöschten Dateien.
- `git add .` fügt nur die neuen und geänderten Dateien hinzu, nicht aber die gelöschten Dateien.

git commit

Mit `git commit` werden die Files aus dem Staging-Bereich in das lokale Repository verschoben. Mit anderen Worten, es speichert die Schnapschüsse der Änderungen, die am lokalen Repository vorgenommen wurden, anstatt das gesamte Repository noch einmal blind zu kopieren. Jedes Mal, wenn Sie eine Übergabe vornehmen, müssen Sie eine kurze Nachricht eingeben, in der Sie die vorgenommenen Änderungen erklären.

Zum Beispiel:

```
git commit -m "das Zwei-Summen-Problem  
gelöst"
```

git pull

Der Befehl `git pull` hilft dabei, Ihr lokales Projektverzeichnis auf dem neuesten Stand zu halten. Er überträgt alle Änderungen, die ein anderer Teilnehmer im entfernten Repository vorgenommen hat, in Ihr lokales Repository.



Es ist eine gute Praxis, die Änderungen vor dem Push-Befehl zu ziehen (`git pull`), wenn Sie an einem Projekt eines Teams arbeiten.

git push

Der Befehl `git push` überträgt die Änderungen aus dem lokalen Repository in das Remote-Repository (z.B. GitHub). So werden die von Ihnen vorgenommenen Änderungen im zentralen Repository veröffentlicht und online verfügbar gemacht. Befehle:

`git push` pusht die Commits in das Remote-Repository

git clone

git clone wird dazu verwendet den Inhalt der auf einem Remote-Repository liegt auf den eigenen Rechner zu klonen. So können Sie beispielsweise eine Codevorlage, die bereits auf Github vorhanden ist, auf ihren Computer kopieren um daran zu arbeiten.

Der Vollständige Befehl lautet z.B.

```
git pull  
https://github.com/bzz-fgict/Java_Basic_Template.git
```



© Kevin Maurizi

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

<https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu06/lu06b-git?rev=1711631267>

Last update: 2024/03/28 14:07

