

Push und Pull in Git – Daten zwischen lokal und remote synchronisieren ☐☐

Wenn du dein lokales Repository mit GitHub (oder einem anderen Remote-Server) verbunden hast, brauchst du eine Möglichkeit, deine Änderungen **hochzuladen** und die Änderungen anderer (oder deine eigenen von einem anderen Gerät) **herunterzuladen**. Genau dafür gibt es **Push** und **Pull** – zwei der wichtigsten Befehle im Git-Alltag.

Das Grundkonzept verstehen

Stell dir dein lokales Repository und das Remote-Repository auf GitHub als zwei getrennte Kopien deines Projekts vor. Sie sind zwar miteinander verbunden, aber **nicht automatisch synchron**. Änderungen, die du lokal machst, existieren zunächst nur auf deinem Computer – und umgekehrt.

flowchart LR

```
LOCAL["☐☐Lokales Repository\nauf deinem Computer"] -->|git push| REMOTE["☐☐ Remote Repository\nauf GitHub"]
REMOTE -->|git pull| LOCAL
```

Befehl	Richtung	Was passiert?
Push	Lokal → Remote	Deine lokalen Commits werden auf GitHub hochgeladen
Pull	Remote → Lokal	Änderungen von GitHub werden auf deinen Computer heruntergeladen und integriert

Push – Deine Änderungen hochladen ☐☐

Wann benutze ich Push?

Du verwendest **Push** immer dann, wenn du:

- Neue Commits gemacht hast, die du auf GitHub sichern möchtest
- Deinen Code von einem anderen Gerät aus zugänglich machen willst
- Anderen Personen (bei Teamarbeit) deine Änderungen zur Verfügung stellen möchtest
- Ein Backup in der Cloud haben willst

☐☐ **Wichtig:** Push lädt nur **Commits** hoch, nicht einfach gespeicherte Dateien. Du musst also erst `git add` und `git commit` gemacht haben, bevor Push etwas zu übertragen hat.

Push in PhpStorm durchführen

1. Stelle sicher, dass du Commits hast, die noch nicht gepusht wurden

In der unteren Statusleiste von PhpStorm siehst du oft einen Hinweis wie „↑2“ – das bedeutet, du hast zwei Commits, die noch nicht auf dem Remote sind.

2. Push-Dialog öffnen

Es gibt mehrere Wege:

- **Tastenkombination:** `Ctrl + Shift + K` (Windows/Linux) oder `Cmd + Shift + K` (macOS)
- **Menü:** Gehe zu **Git → Push...**
- **Toolbar:** Klicke auf den grünen Pfeil nach oben in der Git-Toolbar

3. Commits überprüfen

Im Push-Dialog siehst du eine Liste aller Commits, die hochgeladen werden. Du kannst hier noch einmal prüfen, ob alles korrekt ist.

4. Push ausführen

Klicke auf **Push**. PhpStorm verbindet sich mit GitHub und lädt deine Commits hoch. Bei Erfolg siehst du eine Bestätigungsmeldung unten rechts.

Was kann schiefgehen?

Manchmal verweigert GitHub den Push mit einer Fehlermeldung wie „rejected – non-fast-forward“. Das passiert, wenn auf dem Remote Änderungen existieren, die du lokal noch nicht hast. In diesem Fall musst du **erst Pull ausführen**, bevor du pushen kannst.

Pull – Änderungen herunterladen und integrieren

Wann benutze ich Pull?

Du verwendest **Pull** immer dann, wenn du:

- An mehreren Geräten arbeitest und die Änderungen vom anderen Gerät holen möchtest
- Mit anderen im Team arbeitest und deren Commits integrieren willst
- Vor dem Push sicherstellen möchtest, dass du auf dem neuesten Stand bist
- Nach einer Pause am Projekt weitermachen willst

Was passiert bei einem Pull genau?

Ein `git pull` ist eigentlich eine **Kombination aus zwei Befehlen**:

flowchart LR

```
A["git pull"] --> B["git fetch\nÄnderungen herunterladen"]
B --> C["git merge\nÄnderungen integrieren"]
```

1. **Fetch:** Git lädt die neuen Commits vom Remote herunter, ohne sie direkt anzuwenden
2. **Merge:** Git integriert diese Commits in deinen aktuellen Branch

Pull in PhpStorm durchführen

1. Pull-Dialog öffnen

- **Tastenkombination:** `Ctrl + T` (Windows/Linux) oder `Cmd + T` (macOS)
- **Menü:** Gehe zu **Git** → **Pull...**
- **Toolbar:** Klicke auf den blauen Pfeil nach unten in der Git-Toolbar

2. Optionen prüfen

Im Pull-Dialog siehst du:

- **Remote:** Normalerweise „origin“ (dein GitHub-Repository)

- **Branch:** Der Branch, von dem du pullen möchtest (meist der gleiche wie dein aktueller)
- **Update Type:** Hier kannst du zwischen „Merge“ und „Rebase“ wählen – als Anfänger bleib bei der Standardeinstellung (Merge)

3. Pull ausführen

Klicke auf **Pull**. PhpStorm lädt die Änderungen herunter und integriert sie.

Die schnelle Alternative: Update Project

PhpStorm bietet auch eine komfortablere Option namens **Update Project**:

- **Tastenkombination:** `Ctrl + T` (führt direkt Update aus, wenn so konfiguriert)
- **Menü:** **Git** → **Update Project...**

Diese Option prüft automatisch, ob Änderungen vorhanden sind, und führt den Pull durch. Du wirst gefragt, ob du „Merge“ oder „Rebase“ verwenden möchtest – wähle als Anfänger **Merge**.

Push und Pull im Vergleich – eine Übersicht

Aspekt	Push 	Pull 
Richtung	Lokal → Remote	Remote → Lokal
Zweck	Eigene Commits hochladen	Fremde/andere Commits herunterladen
Tastenkürzel	<code>Ctrl/Cmd + Shift + K</code>	<code>Ctrl/Cmd + T</code>
Voraussetzung	Du musst Commits haben	Es müssen Änderungen auf dem Remote sein
Kann Konflikte verursachen?	Nein (wird ggf. abgelehnt)	Ja (bei gleichzeitigen Änderungen)

Ein typischer Arbeitsablauf in der Praxis

Hier ist ein Beispiel, wie Push und Pull in deinen normalen Workflow passen:

flowchart TD

```
A["☐☐Arbeitstag beginnen"] --> B["Pull ausführen\nNeueste Änderungen holen"]
B --> C["An Dateien arbeiten\nCode schreiben"]
C --> D["Änderungen stagen\nund committen"]
D --> E{"Weitere Arbeit\ngeplant?"}
E -->|Ja| C
E -->|Nein| F["Push ausführen\nÄnderungen hochladen"]
F --> G["☐☐Feierabend"]
```

Merke dir diese Faustregel:

“„Pull am Anfang, Push am Ende“ – Hole dir zu Beginn einer Arbeitssession immer die neuesten Änderungen und lade deine Arbeit am Ende hoch.

Häufige Situationen und wie du damit umgehst

Situation 1: Push wird abgelehnt

Problem: Du versuchst zu pushen, aber Git sagt „rejected“ oder „non-fast-forward“.

Lösung:

1. Führe zuerst einen **Pull** aus
2. Löse eventuelle Merge-Konflikte (falls vorhanden)
3. Versuche den **Push** erneut

Situation 2: Pull verursacht Merge-Konflikte

Problem: Beim Pull meldet PhpStorm, dass es Konflikte gibt.

Lösung: PhpStorm öffnet automatisch den Merge-Tool-Dialog. Dort kannst du für jede konfliktbehaftete Datei entscheiden, welche Version du behalten möchtest – oder beide manuell zusammenführen. (Details dazu findest du im Kapitel zu Merge-Konflikten.)

Situation 3: Du möchtest nur schauen, ob es Änderungen gibt

Problem: Du willst wissen, ob auf GitHub neue Commits sind, ohne sie direkt zu integrieren.

Lösung: Verwende **Fetch** statt Pull:

- Menü: **Git → Fetch**
 - Das lädt die Informationen herunter, ändert aber nichts an deinen lokalen Dateien. Im Git-Log siehst du dann, ob „origin/main“ weiter ist als dein lokaler „main“.
-

Zusammenfassung □

- **Push** = Hochladen: Schickt deine lokalen Commits an das Remote-Repository (z. B. GitHub)
 - **Pull** = Herunterladen + Integrieren: Holt Commits vom Remote und fügt sie in deinen lokalen Branch ein
 - Beide Befehle sind in PhpStorm bequem über Menü, Toolbar oder Tastenkürzel erreichbar
 - **Reihenfolge beachten:** Im Zweifel erst Pull, dann Push – so vermeidest du Ablehnungen
 - Pull kann Merge-Konflikte verursachen, wenn dieselben Stellen geändert wurden – diese lassen sich in PhpStorm komfortabel lösen
-

Revision #1

Created 2026-06-17 03:37:58 UTC by art10m

Updated 2026-06-17 03:39:27 UTC by art10m