

Merge-Konflikte verstehen und in PhpStorm lösen ☐☐

Ein **Merge-Konflikt** ist einer der Momente, vor denen Git-Anfänger oft Respekt haben – aber wenn du einmal verstanden hast, was passiert und wie du damit umgehst, verliert er schnell seinen Schrecken. PhpStorm bietet dir dafür ein hervorragendes visuelles Werkzeug, das die Auflösung deutlich erleichtert.

Was ist ein Merge-Konflikt?

Ein Merge-Konflikt entsteht, wenn Git **nicht automatisch entscheiden kann**, wie zwei unterschiedliche Änderungen zusammengeführt werden sollen. Das passiert, wenn **dieselbe Stelle in einer Datei** in beiden Branches verändert wurde – Git weiß dann nicht, welche Version die „richtige“ ist und bittet dich um Hilfe.

Wann tritt ein Konflikt auf?

Konflikte entstehen typischerweise in diesen Situationen:

- **Zwei Branches bearbeiten dieselbe Codezeile:** Du änderst in deinem Feature-Branch eine Funktion, während jemand (oder du selbst) im Main-Branch dieselbe Zeile ebenfalls angepasst hat.
- **Eine Datei wird in einem Branch gelöscht, im anderen bearbeitet:** Git weiß nicht, ob die Datei weg soll oder die Bearbeitung wichtig ist.
- **Parallele Arbeit an derselben Datei:** Selbst wenn du alleine arbeitest, kann das passieren, wenn du z. B. einen Hotfix direkt im Main-Branch machst, während du parallel an einem Feature arbeitest.

Wann gibt es *keinen* Konflikt?

Wenn die Änderungen in **unterschiedlichen Dateien** oder in **unterschiedlichen Bereichen derselben Datei** stattfinden, kann Git diese problemlos automatisch zusammenführen – kein Konflikt.

Ein konkretes Beispiel

Stell dir vor, du hast eine Datei `config.php` mit folgender Zeile:

```
$siteTitle = "Meine Website";
```

Jetzt passiert Folgendes:

1. Im **Main-Branch** änderst du die Zeile zu:

```
$siteTitle = "Meine tolle Website";
```

2. Im **Feature-Branch** (den du vorher vom Main-Branch abgezweigt hast) änderst du dieselbe Zeile zu:

```
$siteTitle = "Mein Webprojekt";
```

Wenn du jetzt versuchst, den Feature-Branch in den Main-Branch zu mergen, hat Git ein Problem: Beide Branches haben **dieselbe Zeile unterschiedlich verändert**. Git kann nicht wissen, ob du „Meine tolle Website“ oder „Mein Webprojekt“ haben möchtest – oder vielleicht sogar etwas ganz anderes.

flowchart TD

```
A["Ursprung<br>$siteTitle = Meine Website"] --> B["Main-Branch<br>$siteTitle = Meine tolle Website"]
A --> C["Feature-Branch<br>$siteTitle = Mein Webprojekt"]
B --> D["Merge-Versuch"]
C --> D
D --> E["Konflikt!<br>Git weiss nicht,<br>welche Version gilt"]
```

Merge-Konflikt in PhpStorm lösen – Schritt für Schritt

Angenommen, du bist im **Main-Branch** und willst deinen Feature-Branch mergen:

1. **Merge starten**

Klicke unten rechts auf den Branch-Namen (z. B. „main“), wähle deinen Feature-Branch aus der Liste und klicke auf **Merge into Current**. Alternativ: **Git → Merge...** im Menü.

2. Konflikt-Meldung erscheint

Wenn ein Konflikt existiert, zeigt PhpStorm dir ein Dialogfenster mit dem Titel **Merge Conflicts**. Hier siehst du eine Liste aller Dateien, die Konflikte haben – in unserem Beispiel `config.php`.

3. Konflikt-Editor öffnen

Wähle die konfliktbehaftete Datei aus und klicke auf **Merge...** (oder doppelklicke auf die Datei). Es öffnet sich der **3-Wege-Merge-Editor** – das Herzstück der Konfliktlösung in PhpStorm.

4. Den 3-Wege-Editor verstehen

Du siehst drei Spalten:

Linke Spalte	Mittlere Spalte	Rechte Spalte
Dein aktueller Branch (Main)	Das Ergebnis (was am Ende gespeichert wird)	Der eingehende Branch (Feature)

Die konfliktbehafteten Stellen sind farblich hervorgehoben. In unserem Beispiel siehst du links „Meine tolle Website“ und rechts „Mein Webprojekt“.

5. Änderungen übernehmen oder kombinieren

Für jede konfliktbehaftete Stelle hast du mehrere Möglichkeiten:

- **Accept Left** (Pfeil-Button neben der linken Version): Übernimmt die Version aus dem Main-Branch.
- **Accept Right** (Pfeil-Button neben der rechten Version): Übernimmt die Version aus dem Feature-Branch.
- **Accept Both**: Übernimmt beide Versionen nacheinander (selten sinnvoll bei derselben Zeile, aber nützlich bei Codeblöcken).
- **Manuell bearbeiten**: Du kannst in der mittleren Spalte direkt den Text anpassen und etwas völlig Neues schreiben, z. B.:

```
$siteTitle = "Mein tolles Webprojekt";
```

6. Alle Konflikte lösen

Gehe jeden markierten Konflikt durch und triff eine Entscheidung. Die mittlere Spalte zeigt dir immer das aktuelle Ergebnis. Sobald alle Konflikte gelöst sind (keine roten Markierungen mehr), wird der **Apply**-Button aktiv.

7. Änderungen übernehmen

Klicke auf **Apply**. Der Konflikt für diese Datei ist gelöst. Wenn es mehrere Dateien mit Konflikten gibt, wiederhole den Prozess für jede.

8. Merge abschließen

Nachdem alle Konflikte gelöst sind, erstellt PhpStorm automatisch einen **Merge-Commit**. Du kannst die vorgeschlagene Commit-Nachricht anpassen (z. B. „Merge branch ‚feature-xyz‘ into main“) und den Commit bestätigen.

Tipps für den Umgang mit Konflikten

- **Keine Panik!** Ein Konflikt bedeutet nicht, dass etwas kaputt ist – Git bittet dich nur um eine Entscheidung, die es selbst nicht treffen kann.
 - **Verstehe beide Versionen:** Bevor du eine Version wählst, schau dir an, *warum* die Änderung gemacht wurde. Manchmal ist die beste Lösung eine Kombination aus beiden.
 - **Teste nach dem Merge:** Nachdem du Konflikte gelöst hast, teste dein Projekt gründlich. Syntaktisch kann alles korrekt aussehen, aber logisch kann etwas nicht zusammenpassen.
 - **Konflikte vermeiden:** Je öfter du deine Branches synchronisierst (regelmäßig `main` in deinen Feature-Branch mergen), desto kleiner und seltener werden Konflikte.
 - **Im Zweifel: Abbrechen:** Wenn du dir unsicher bist, kannst du den Merge jederzeit mit **Git → Abort Merge** abbrechen. Alles bleibt beim Alten, und du kannst dich in Ruhe vorbereiten.
-

Zusammenfassung

Ein Merge-Konflikt entsteht, wenn Git zwei unterschiedliche Änderungen an derselben Stelle nicht automatisch zusammenführen kann. Das ist kein Fehler, sondern eine **Aufforderung zur Entscheidung**. PhpStorm bietet dir mit dem 3-Wege-Merge-Editor ein mächtiges Werkzeug, um diese Konflikte visuell und komfortabel zu lösen – du siehst beide Versionen nebeneinander, kannst gezielt auswählen oder kombinieren und behältst immer die volle Kontrolle über das Ergebnis.

Revision #2

Created 2026-06-17 01:44:31 UTC by art10m

Updated 2026-06-17 07:05:52 UTC by art10m