

Die Git-Historie in PhpStorm anzeigen und verstehen

Nachdem du einige Commits gemacht hast, möchtest du natürlich auch **nachschauen können**, was du wann geändert hast. PhpStorm bietet dafür ein mächtiges, visuell ansprechendes Werkzeug: das **Git Log**. Hier erfährst du, wie du es nutzt und die Informationen richtig interpretierst.

Das Git-Log-Fenster öffnen

Es gibt mehrere Wege, um zur Commit-Historie zu gelangen:

1. **Über das Menü:** Gehe zu **Git → Show Git Log** (oder in älteren Versionen **VCS → Git → Show History**).
2. **Über das Tool-Fenster:** Am unteren Rand von PhpStorm findest du den Reiter **Git**. Klicke darauf, und du siehst automatisch den **Log**-Tab.
3. **Tastenkombination:** Mit `Alt + 9` (Windows/Linux) bzw. `Cmd + 9` (macOS) öffnest du das Git-Tool-Fenster direkt.

Das Log-Fenster zeigt dir eine **chronologische Liste aller Commits** – die neuesten oben, die ältesten unten.

Die Ansicht verstehen

Das Git-Log in PhpStorm ist in mehrere Bereiche aufgeteilt:

```
graph TD
    subgraph LOG ["Git Log Fenster"]
        A["Commit-Liste\nalle Commits chronologisch"] --> B["Commit-Details\nNachricht, Autor, Datum"]
        B --> C["Geänderte Dateien\nListe der betroffenen Files"]
        C --> D["Diff-Ansicht\nkonkrete Änderungen im Code"]
    end
```

Die Commit-Liste (linker/oberer Bereich)

Hier siehst du für jeden Commit:

- **Commit-Nachricht** – die erste Zeile deiner Nachricht (deshalb ist eine aussagekräftige erste Zeile so wichtig!)
- **Autor** – wer den Commit gemacht hat
- **Datum und Uhrzeit** – wann der Commit erstellt wurde
- **Commit-Hash** – eine eindeutige ID (z. B. `a3b8f2c`), mit der du den Commit referenzieren kannst
- **Branch-Tags** – farbige Labels zeigen an, zu welchem Branch ein Commit gehört

💡 **Tipp:** Die grafische Darstellung links neben den Commits zeigt dir die **Branch-Struktur** – also wo Branches abzweigen und wieder zusammengeführt werden.

Die Commit-Details (mittlerer Bereich)

Wenn du einen Commit in der Liste anklickst, siehst du rechts daneben oder darunter:

- Die **vollständige Commit-Nachricht** (auch mehrzeilige)
- Den **vollständigen Commit-Hash**
- Informationen über den **Autor** und den **Committer** (können unterschiedlich sein)

Die geänderten Dateien

Unterhalb der Commit-Details findest du eine **Liste aller Dateien**, die in diesem Commit geändert wurden. Jede Datei hat ein Symbol, das die Art der Änderung anzeigt:

Symbol/Farbe	Bedeutung
□ Grün / +	Datei wurde neu hinzugefügt
□ Blau / ~	Datei wurde geändert
□ Rot / -	Datei wurde gelöscht
□ Lila	Datei wurde umbenannt oder verschoben

Die konkreten Änderungen ansehen (Diff)

Das eigentlich Spannende ist natürlich: **Was genau wurde geändert?** Dafür nutzt du die Diff-Ansicht:

1. **Datei auswählen:** Klicke in der Liste der geänderten Dateien auf eine Datei.
2. **Diff öffnen:** Doppelklicke auf die Datei oder drücke `Strg + D` (Windows/Linux) bzw. `Cmd + D` (macOS).
3. **Änderungen lesen:** PhpStorm zeigt dir eine **Side-by-Side-Ansicht**:
 - **Links:** Der alte Stand (vor dem Commit)
 - **Rechts:** Der neue Stand (nach dem Commit)

Gelöschte Zeilen sind rot markiert, hinzugefügte grün. So siehst du auf einen Blick, was sich geändert hat.

“ **Tipp:** Du kannst die Ansicht auch auf „Unified“ umstellen, wenn du lieber alles untereinander statt nebeneinander sehen möchtest. Das geht über das Zahnrad-Symbol in der Diff-Ansicht.

Nützliche Filter und Suchfunktionen

Bei größeren Projekten wird die Historie schnell lang. PhpStorm bietet dir praktische **Filter**, um den Überblick zu behalten:

- **Nach Text suchen:** Gib im Suchfeld einen Begriff ein, um Commits zu finden, deren Nachricht diesen Begriff enthält.
 - **Nach Autor filtern:** Klicke auf das Filter-Symbol und wähle „User“, um nur Commits eines bestimmten Autors zu sehen.
 - **Nach Datum filtern:** Du kannst auch einen Zeitraum eingrenzen, z. B. „letzte Woche“ oder ein bestimmtes Datum.
 - **Nach Pfad filtern:** Besonders praktisch – du kannst die Historie auf eine **einzelne Datei oder einen Ordner** beschränken. Rechtsklicke dafür im Projektbaum auf eine Datei und wähle **Git → Show History for Selection**.
-

Historie einer einzelnen Datei anzeigen

Manchmal interessiert dich nicht die gesamte Projekt-Historie, sondern nur: **Was ist mit dieser einen Datei passiert?**

1. Rechtsklicke auf die Datei im Projektbaum.
2. Wähle **Git → Show History**.
3. Du siehst jetzt nur die Commits, die **diese Datei betreffen**.

Noch detaillierter wird es mit **Annotate** (auch „Blame“ genannt):

1. Öffne die Datei im Editor.
2. Rechtsklicke in den linken Rand (wo die Zeilennummern stehen).
3. Wähle **Annotate with Git Blame**.

Jetzt siehst du **für jede Zeile**, in welchem Commit sie zuletzt geändert wurde, von wem und wann. Das ist unglaublich praktisch, um herauszufinden, wann und warum eine bestimmte Code-Stelle entstanden ist.

Praktisches Beispiel

Angenommen, du hast in den letzten Tagen mehrere Commits gemacht und möchtest nachschauen, wann du die Datei `database.php` zuletzt geändert hast:

1. Öffne das Git-Log mit `Alt + 9`.
2. Gib im Suchfeld oben `database.php` ein oder nutze den Pfad-Filter.
3. Du siehst jetzt alle Commits, die diese Datei betreffen.
4. Klicke auf einen Commit und dann doppelt auf `database.php` in der Dateiliste.
5. Die Diff-Ansicht zeigt dir genau, welche Zeilen du damals geändert hast.

Zusammenfassung

Aktion	Weg in PhpStorm
Gesamte Historie anzeigen	Git → Show Git Log oder <code>Alt + 9</code>

Aktion	Weg in PhpStorm
Änderungen eines Commits sehen	Commit anklicken → Datei doppelklicken
Historie einer Datei	Rechtsklick → Git → Show History
Wer hat welche Zeile geschrieben?	Rechtsklick im Editor → Annotate with Git Blame
Nach Commits suchen	Suchfeld im Log-Fenster nutzen

Mit diesen Werkzeugen hast du deine Projektgeschichte immer im Griff – du kannst jederzeit nachvollziehen, *was sich wann und warum* geändert hat. ☐☐

Revision #1

Created 2026-06-16 16:24:51 UTC by art10m

Updated 2026-06-16 16:25:49 UTC by art10m