

# Ein Projekt unter Versionskontrolle stellen

Bevor Commits, Branches oder Pull Requests überhaupt möglich sind, muss PhpStorm wissen, dass ein Projektverzeichnis von Git verwaltet werden soll. Technisch entspricht dieser Schritt exakt dem Kommandozeilenbefehl `git init`, wird in PhpStorm jedoch über einen geführten Dialog ausgeführt, der zusätzlich die *VCS-Zuordnung* (Directory Mapping) im Projekt hinterlegt. Diese Zuordnung ist der Grund, warum die IDE anschließend Dateifarben, Änderungslisten und das Commit-Fenster überhaupt anzeigen kann.

## Zwei Ausgangssituationen

In der Praxis triffst du in PhpStorm auf zwei unterschiedliche Startpunkte:

1. **Neues, noch unversioniertes Projekt** – ein lokaler Ordner mit PHP-Dateien, aber ohne `.git`-Verzeichnis.
2. **Bereits vorhandenes Repository** – das Projekt wurde per `git clone` oder außerhalb von PhpStorm erstellt und muss nur noch korrekt zugeordnet werden.

Dieses Unterkapitel behandelt den ersten Fall ausführlich, da er den eigentlichen Einrichtungsschritt darstellt.

## Schritt für Schritt: Version Control Integration aktivieren

1. Öffne das Projekt in PhpStorm.
2. Wähle im Menü **VCS → Enable Version Control Integration...**
3. Im erscheinenden Dialog wählst du als System **Git** aus.
4. Bestätige mit **OK**.

PhpStorm führt daraufhin im Hintergrund `git init` im Projektstammverzeichnis aus und erstellt das `.git`-Verzeichnis. Gleichzeitig trägt die IDE das Projektverzeichnis als *VCS Root* ein – sichtbar unter:

“ **Praxistipp:** Ist das VCS-Menü nicht sichtbar, wurde entweder kein Projekt geöffnet oder es liegt bereits eine Zuordnung vor. In diesem Fall erscheint statt *Enable Version Control Integration* direkt das erweiterte VCS-Menü mit Commit-, Branch- und Log-Optionen.

# Was sich im Projektbaum ändert

Unmittelbar nach der Aktivierung markiert PhpStorm alle Dateien im *Project*-Werkzeugfenster farblich:

| Farbe        | Bedeutung                                                        |
|--------------|------------------------------------------------------------------|
| Braun/Orange | Datei ist <i>unversioniert</i> (noch nicht zu Git hinzugefügt)   |
| Grün         | Datei ist neu hinzugefügt und für den nächsten Commit vorgemerkt |
| Blau         | Datei ist bereits verfolgt und wurde geändert                    |
| Grau         | Datei ist ignoriert (siehe <code>.gitignore</code> )             |

Da direkt nach `git init` noch nichts vorgemerkt ist, erscheinen zunächst *alle* Dateien braun. Dies entspricht dem Zustand, den `git status` auf der Kommandozeile als „Untracked files“ ausweist.

## Dateien zur Versionskontrolle hinzufügen

PhpStorm bietet nach der Aktivierung häufig automatisch einen Hinweis an:

“ „The following files are not currently tracked by Git...”

Über **Add** werden die ausgewählten Dateien in die Staging Area übernommen – identisch zu `git add`. Alternativ erreichst du dies jederzeit über:

- Rechtsklick auf Datei oder Ordner → **Git** → **Add**

- Tastenkürzel Ctrl + Alt + A

Vor diesem Schritt lohnt es sich, kurz zu prüfen, ob bereits eine `.gitignore`-Datei existiert, damit keine Abhängigkeiten wie `vendor/` oder IDE-Konfigurationsdateien versehentlich mit aufgenommen werden. Eine vertiefte Behandlung dieses Themas folgt im Kapitel „Dateien ignorieren und Repository-Hygiene“.

## Der Ablauf im Überblick

flowchart LR

```
A["Projektordner ohne .git"] --> B["VCS -> Enable Version Control Integration"]
B --> C["git init im Hintergrund"]
C --> D["Directory Mapping wird angelegt"]
D --> E["Dateien erscheinen als unversioniert"]
E --> F["Dateien ueber Add vormerken"]
F --> G["Bereit fuer ersten Commit"]
```

```
style A fill:#374151,color:#ffffff
```

```
style G fill:#065f46,color:#ffffff
```

## Directory Mappings verstehen

Gerade bei PHP-Projekten mit mehreren Modulen, einer separaten `docs`-Struktur oder eingebundenen Bibliotheken ist es wichtig zu wissen, *welches* Verzeichnis PhpStorm als Wurzel des Repositories betrachtet. Unter

Settings/Preferences → Version Control → Directory Mappings

siehst du die Zuordnung von Dateisystempfad zu VCS-System. In den allermeisten Fällen genügt ein einziger Eintrag mit dem Projektstammverzeichnis. Enthält dein Projekt jedoch mehrere unabhängige Repositories – etwa ein separates Repository für ein Composer-Paket innerhalb desselben PhpStorm-Projekts – kannst du hier zusätzliche Mappings ergänzen.

⚠ **Warnung:** Verschachtelte `.git`-Verzeichnisse (ein Repository innerhalb eines anderen) führen ohne Submodule oder Subtree-Konfiguration zu Verwirrung, da Git das innere Repository ignoriert, PhpStorm es aber unter Umständen separat

anzeigt. Vermeide dies, solange du die Konzepte aus Kapitel 27 noch nicht behandelt hast.

# Vergleich: IDE-Aktion und Git-Befehl

| PhpStorm-Aktion                          | Entsprechender Git-Befehl                  |
|------------------------------------------|--------------------------------------------|
| VCS → Enable Version Control Integration | <code>git init</code>                      |
| Rechtsklick → Git → Add                  | <code>git add &lt;Datei&gt;</code>         |
| Directory Mappings anzeigen              | <code>git rev-parse --show-toplevel</code> |
| Farbliche Statusanzeige im Projektbaum   | <code>git status</code>                    |

Diese Gegenüberstellung ist bewusst Teil jedes praktischen Kapitels dieses Kurses: Sie stellt sicher, dass du auch dann souverän bleibst, wenn kein grafisches Werkzeug zur Verfügung steht – etwa bei der Arbeit über SSH auf einem Server.

## Kontrolle des Ergebnisses

Nach erfolgreicher Aktivierung solltest du folgende Punkte überprüfen:

- Im Projektverzeichnis existiert ein `.git`-Ordner (sichtbar über den Datei-Explorer bei aktivierter Anzeige versteckter Dateien).
- Das VCS-Menü in PhpStorm zeigt nun vollständige Optionen wie **Commit**, **Push** und **Branches** an.
- Das *Commit*-Werkzeugfenster listet die vorgemerkten Dateien korrekt auf.

Damit ist das Fundament gelegt: Das Projekt befindet sich nun vollständig unter Versionskontrolle, und die IDE ist bereit, den ersten Commit entgegenzunehmen – der Gegenstand des folgenden Unterkapitels zur Konfiguration des Commit-Werkzeugfensters. ☐

Revision #1

Created 2026-07-26 17:06:01 UTC by art10m

Updated 2026-07-26 17:06:43 UTC by art10m