

Plugin-Verwaltung

Strategien zu Auswahl, Verwaltung, Konfiguration und Aktualisierung von Plugins

- [Plugin-Aktualisierung](#)

Plugin-Aktualisierung

Hier wird beschrieben, wie Plugins bereitgestellt werden und worauf bei der Aktualisierung zu achten ist.

Einleitender Überblick

Plugins werden auf unterschiedlichste Weise bereitgestellt. Unterschiede liegen hierbei in Update-Regelmäßigkeit, Code-Qualität, Versions-Struktur und Repository-Nutzung. Am häufigsten wird github als Repository genutzt. Darüber hinaus sind die meisten beliebten Plugins auch im offiziellen [moodle.org Plugin-Verzeichnis](#) zu finden. Dies ist die erste und Standard-Anlaufstelle für die Suche nach Plugins. Hier veröffentlichte Plugins mussten bei der Erstveröffentlichung eine Basis-Qualitätssicherung vom Moodle Headquater (HQ) durchlaufen ([Plugin contribution guidelines](#)).

Repository-Aufbau und Dateistruktur

Ein Repository, z.B. auf github, enthält die Plugin-Dateien (i. d. R. PHP-Code) mit den Infos zu Veröffentlichungsdatum und Änderungsverlauf. Änderungen werden als sogenannte "commits" mit "commit"-messages veröffentlicht, die im besten Falle eine hilfreiche Beschreibung der gemachten Änderungen enthalten, z.B. "Release v5.1" oder "Bugfix: issue XYZ". Für die Installation von Plugins sind diese beiden Dateien am wichtigsten:

- *README.md*
- *version.php*

Der Inhalt der README wird direkt unterhalb der Dateien angezeigt und enthält Infos, wie z.B. zu Installation und Konfiguration.

Die *version.php* enthält, neben der offensichtlichen Versionsnummer, Release-Bezeichnung, minimal benötigte Moodle-Version (*requires*), unterstützte Moodle-Version von bis (*supported* - "[405, 501]" bedeutet z.B. 4.5 bis 5.1. unterstützt), techn. Plugin-Namen (*component*), Reifegrad (maturity) und optional Abhängigkeiten von anderen Plugins (*dependencies*). Versionsnummern für Moodle werden als 10-stellige Zahl mit vorangestellter Jahreszahl angegeben, z.B. 2025100600 für Moodle 5.1. Welche Version hinter den Zahlen steckt, kann unter <https://moodledev.io/general/releases> nachgesehen werden.

Branches und Versionierung

Jedes Repository hat einen Standard-branch, der nach Aufruf der Startseite des Repositorys angezeigt wird. Meist heißt dieser "main", ehemals war die Standardbezeichnung "master".

Viele Repositories arbeiten mit zusätzlichen branches. Diese enthalten entweder angelehnt an die moodle-core-Dateien die für die jeweilige Moodle-Version passenden Plugin-Versionen, z.B. *MOODLE_405* für Moodle 4.5, *MOODLE_500* für Moodle 5.0 oder *MOODLE_501* für Moodle 5.1. Oder es können auch Entwicklungs-branches, wie z.B. *DEV*, oder branches für gewisse neue Funktionen, z.B. *Feature-XYZ*, sein.

Das Schwierige ist, dass es hierbei **keinen allgemeingültigen Standard** gibt, an den sich alle Entwickelnden halten. **Es muss also bei Installation und Updates von Plugins aus git-Repositories herausgefunden werden, ob die für die benötigte Moodle-Version richtigen Plugin-Dateien im branch main oder einem anderen liegen.** Im Zweifelsfall gibt die *version.php* Auskunft, welche Versionen unterstützt werden.

moodle.org Plugin-Verzeichnis

Etwas einfacher ist es bei Nutzung des moodle.org Plugin-Verzeichnisses. Hier werden im Tab "Versions" die verfügbaren Versionen des Plugins gezeigt inkl. der unterstützten Moodle-Versionen. Wenn das eigene Moodle-System für den moodle.org Account eingerichtet und ein Single-Webserver ist, kann sogar direkt via Button "Install Now" ein Plugin darauf installiert werden. Bei mehreren Webservern hingegen ist das Vorgehen über eine Frontend-Installation tückisch, da die Plugin-Dateien nur auf den aktuellen Webserver kopiert werden, nicht jedoch auf den/die anderen. Alternativ können die Plugin-Dateien heruntergeladen und manuell ins passende Webserver-Verzeichnis kopiert werden, z.B. */theme/boost_union* für Boost Union; die meisten Plugins geben den nötigen Pfad in der Readme an.

Benachrichtigung über Plugin-Updates

In der *Website-Administration* unter "*Plugins->Plugin-Übersicht*" werden nur für auf moodle.org geführte Plugins *Verfügbare Aktualisierungen* angezeigt. Nach gleichem Prinzip werden auch nur für moodle.org Plugins Mails mit "Für mindestens ein Plugin ist eine Aktualisierung verfügbar" an die im System hinterlegte admin-Adresse versandt. **Für Plugins, die nicht auf moodle.org registriert sind, ist händisch zu prüfen, ob es Aktualisierungen gibt.**

git zur Plugin-Aktualisierung

Vor allem größere Moodle-Installationen nutzen git-Mechanismen zur Plugin-Verwaltung und -aktualisierung. In Skripten oder z.B. Ansible playbooks wird definiert, welche Plugins mit welchen Versionen oder branches geladen werden sollen und damit das gewünschte Plugin-Set für die Moodle-Instanz orchestriert. Wenn die Repositories sauber auf den Webservern initialisiert sind (i.d.R. via *git submodule add*), können direkt alle git-Befehle dort angewandt werden. Somit kann direkt auf Webserver-Ebene innerhalb des Plugin-Verzeichnisses schnell auf andere Plugin-Versionen gewechselt werden, ohne alle Moodle-Dateien neu generieren zu müssen. Zwei Beispiele zeigen, wie dies genutzt werden kann.

branch-Wechsel für Moodle 5.1

```
git checkout MOODLE_501_STABLE
```

Wechsel auf bestimmtes commit

```
git fetch & git reset --hard 38c0040a0e8fec0e434cf57a8be363d98f11591e
```

Der hier im Beispiel genannte aus Ziffern und Buchstaben bestehende hash lässt sich aus der Web-Adresse des gewünschten commits kopieren, z.B. nach Aufruf von /commit/[branch] (Beispiel:

https://github.com/moodle-an-hochschulen/moodle-theme_boost_union/commits/MOODLE_501_STABLE/).

to-do: Eigene Seite zu git erstellen und hier darauf verweisen

Plugin-Version zurücksetzen (Downgrade-Hack)

Es gibt Szenarien, in denen bewusst auf eine ältere Plugin-Version gewechselt werden soll, z.B. auf Test-Systemen. Dies funktioniert allerdings nicht einfach so, da der Upgrade-Mechanismus keine niedrigeren Versionsnummern zulässt. Bei Änderung einer Plugin-Version auf eine ältere als zuvor installiert, liefert die Upgrade-Prüfung den Fehler: *"Höhere Version ist bereits installiert!"*

Nur durch Eingriff in die Datenbank (DB) kann auf eine ältere Plugin-Version gewechselt werden. Dazu ist in der DB-Tabelle *"mdl_config_plugins"* nach dem gewünschten *"plugin"* zu suchen und die *"version"* auf den älteren Versionswert zu ändern, der zu den eingespielten älteren Plugin-Dateien passt. *Warnung: Dieser Hack kann zu Komplikationen führen, z.B. wenn sich Tabellenstrukturen zwischen den beiden Versionen geändert haben, er ist explizit nur auf Entwicklungs- und Test-Systemen anzuwenden!*

Autor: [Klaus Steitz](#), Technische Universität Darmstadt