

Classroom 50: Noten nach Moodle übertragen

Der Übertrag läuft **zentral** aus dem `classroom50-Config-Repo` und **nachts**, nicht bei jedem Push.

Warum zentral und nachts

Der Moodle-Webservice-Token kann Noten für **beliebige Personen** setzen. In einem Studi-Repo liest ihn ein selbst hinzugefügter Workflow in drei Zeilen aus. Genau so lief es unter GitHub Classroom (`secrets: inherit in classroom.yml`) — diese Exposition endet mit der Migration.

Technisch geht es unter Classroom 50 auch gar nicht anders: Der zentrale Reusable Workflow `autograde-runner.yml` deklariert nur `outputs:` und reicht **keine Secrets** an den Bewertungs-Job durch.

Nachts statt bei jedem Push, weil der direkte Weg einen Token in jedem Studi-Repo, eine eigene Workflow-Datei je Repo, eine nachgebaute Skip-Logik und rund 33 % mehr Actions-Minuten gekostet hätte — und ein Nacht-Fallback wäre trotzdem nötig gewesen.

Voraussetzungen auf der Moodle-Seite

1. Eine Aktivität vom Typ **External Assignment** (`mod_externalassignment`).
2. Ihr Name ist **exakt der Slug** der Aufgabe, z.B. `m323-lu01-a02-imperativer-ggt`.
3. Der **GitHub-Login steht im Moodle-Profil** der lernenden Person, im Feld *GitHub Classroom* — siehe [GitHub Benutzernamen in Moodle eintragen](#).
4. Die Person ist im Kurs eingeschrieben.

Welches Profelfeld gelesen wird, steht in der Plugin-Einstellung `mod_externalassignment/external_username` (Vorgabe: Kurzname `github_user`). Das Plugin sucht ausschliesslich in diesem Feld — der Moodle-Benutzername ist für die Zuordnung **irrelevant**.

Zwei Fehlermeldungen, die man auseinanderhalten muss:

Meldung	Bedeutung
No Moodle user found with username „X“: Update your Moodle profile.	Profelfeld leer oder falsch geschrieben
No matching assignment found. Contact your teacher.	Aktivität fehlt oder die Person ist nicht im Kurs — das Plugin löst (<code>assignmentname</code> , <code>userid</code>) gemeinsam auf

Zum Testen deshalb einen echten Studi-Login nehmen — der eigene Lehrer-Account ist im Kurs meist kein Teilnehmer.

Zugangsdaten

Im **classroom50-Repo**, nicht in der Studi-Organisation:

Art	Name	Wert
Secret	MOODLE_TOKEN	Webservice-Token
Variable	MOODLE_URL	z.B. https://moodle.bzz.ch
Variable	MOODLE_FUNCTION	optional, Vorgabe mod_externalassignment_update_grade

```
gh secret set MOODLE_TOKEN --repo <ORG>/classroom50
gh variable set MOODLE_URL --repo <ORG>/classroom50 --body
"https://moodle.bzz.ch"
```

Für den Token einen **eigenen Webservice-Benutzer** anlegen, der nur diese eine Funktion darf. Je enger die Rechte, desto kleiner der Schaden bei einem Leck.

Was übertragen wird

Je Kombination aus Aufgabe und Person geht die **neueste** Abgabe raus.

Moodle-Feld	Quelle
assignment_name	Slug der Aufgabe aus scores.json
user_name	GitHub-Login der besitzenden Person
points / max	Punkte der neuesten Abgabe
externallink	URL des GitHub-Releases
feedback	Release-Text; davor ein Hinweis bei verspäteter Abgabe, dahinter der Link zur Abgabe

Zeitplan

Job	Zeit	Was er tut
collect-scores	nachts	sammelt alle Releases in <CLASSROOM>/scores.json
moodle-sync	57 4 * * * (04:57)	überträgt daraus nach Moodle

40 Minuten Abstand, damit collect-scores sicher durch ist.

Ein Zustandsfile <CLASSROOM>/moodle-state.json merkt sich, was schon übertragen wurde. Übersprungen wird nur, wenn Abgabe **und** Punktzahl identisch sind — eine Nachbewertung derselben Abgabe geht also erneut raus.

Von Hand auslösen

Im classroom50-Repo unter **Actions → Moodle Sync → Run workflow**. Eingaben: Klassenzimmer, optional Aufgabe und Login, dazu die Schalter dry_run und force. Bleibt das Klassenzimmer leer, laufen **alle** Ordner mit einer scores.json.

```
gh workflow run moodle-sync.yaml --repo <ORG>/classroom50 \
-f classroom=<CLASSROOM> -f dry_run=true
```

Trockenlauf-Ausgabe:

```
Moodle-Übertrag: 2 Abgaben
[dry-run] <SLUG> / anna: 5/7
[dry-run] <SLUG> / bruno: 7/7
übertragen: 2 | unverändert: 0 | fehlgeschlagen: 0
```

Lokal prüfen, ohne Zugangsdaten

```
gh api repos/<ORG>/classroom50/contents/<CLASSROOM>/scores.json \
-H 'Accept: application/vnd.github.raw' > scores.json
GH_TOKEN=$(gh auth token) python -m pygrader50.moodle scores.json --dry-run
```

Alle Optionen: [CLI-Referenz](#).

Probleme

Meldung / Symptom	Ursache
No matching assignment found	Aktivitätsname oder Kursmitgliedschaft passt nicht
keine XML-Antwort erhalten	falsche MOODLE_URL oder ungültiger Token — Moodle liefert eine Login-Seite
Note kommt nicht an, Log sagt „unverändert,“	Zustandsfile hält sie für erledigt — mit force erneut auslösen
Nachlauf jede Nacht rot, immer dieselbe Person	Karteileichen-Eintrag in scores.json, siehe Betriebsfallen

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/howto/git/classroom50/moodle?rev=1787048451>

Last update: **2026/08/18 12:20**

