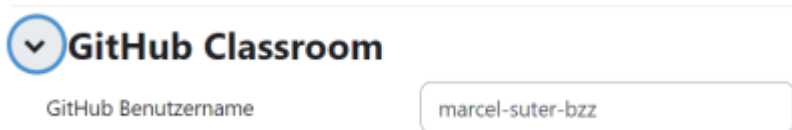


Bewertung vom Autograding in Moodle übernehmen

Voraussetzungen

GitHub Benutzername im Moodle-Profil

Die Lernenden müssen ihren GitHub-Benutzernamen in ihrem Moodle-Profil eintragen. Unter „GitHub Classroom“ befindet sich das Feld „GitHub Benutzername“.



Workflow in Template

1. Erstelle eine Kopie des Repositorys „BZZ-Common/pyhton-template“ um alle benötigten Dateien zu erhalten:
 - `.github/workflows/classroom.yml`
 - `.github/workflows/copyissues.yml` (wird für das automatische Bewerten nicht benötigt)
 - `.github/classroom/autograding.json`
2. Alternativ kannst du die Dateien/Ordner auch manuell erstellen:
 1. Öffne das Repository mit dem Template für das Assignment.
 2. Erstelle die Ordner `.github/classroom` und `.github/workflows`.
 3. Erstelle die Datei `classroom.yml` in `.github/workflows`.

```
name: GitHub Classroom Workflow

on: push

permissions:
  checks: write
  issues: write
  actions: read
  contents: read

jobs:
  copy-issues:
    if: ${ contains(github.actor, 'classroom') }
    uses: BZZ-Commons/workflows/.github/workflows/setup.yml@main
    with:
      SOURCE_REPO: templates-python/m319_lu04_a00_classroom #
      CHANGE owner/name of the repo with the issues
```

```

    TARGET_REPO: ${github.repository}}
    ADD_LABELS: "false"
  secrets:
    GHSECRET: ${secrets.GITHUB_TOKEN}}

  grading:
    if: ${!contains(github.actor, 'classroom')}
    uses: BZZ-Commons/workflows/.github/workflows/grading.yml@main
    with:
      MOODLE: false # CHANGE true=export grade to moodle
    secrets: inherit

```

4. Erstelle die Datei `autograding.json` in `.github/classroom`.
5. Füge deine Testfälle in diese Datei ein.

Beispiel für Testfälle

```

{
  "tests": [
    {
      "name": "test_main1",
      "setup": "sudo -H pip3 install -r requirements.txt",
      "run": "pytest -v -k \"test_main1\"",
      "input": "",
      "output": "",
      "comparison": "included",
      "timeout": 10,
      "points": 1
    },
    {
      "name": "test_main2",
      "setup": "sudo -H pip3 install -r requirements.txt",
      "run": "pytest -v -k \"test_main2\"",
      "input": "",
      "output": "",
      "comparison": "included",
      "timeout": 10,
      "points": 1
    }
  ]
}

```

Classroom Assignment

1. Erstelle ein neues Assignment in GitHub Classroom.
 - Der Name des Assignments wird in Kleinbuchstaben und snake_case geschrieben.
 - GitHub Classroom zeigt den Prefix an, welcher für die Repositories der Lernenden verwendet wird: Zum Beispiel „m319-lu08-a01-multiply“.

- Wähle das Template aus dem letzten Abschnitt aus.



Der Prefix für die Repos ist nicht immer 1:1 der Name des Classrooms. Aktuell (Juli 2023) werden offenbar Underscores durch Bindestriche ersetzt.

Aufgabe in Moodle

1. Erstelle eine Aufgabe in Moodle.
2. Trage den Namen des Classroom Assignments unter „GitHub Classroom“ ⇒ „Classroom Assignment“ ein.

▼ GitHub Classroom

Classroom Assignment

Der Name des GitHub Classroom Assignments zu dieser Aktivität

TODO Das Assignment sollte wie das Repo Kleingeschrieben werden.

Technische Doku

Activity modules

Anzeige	Short name	fieldid
Classroom Assignment	classroom_assignment	2
Classroom Template	classroom_template	?
GitHub Template	gh_template	?
Classroom Link	classroom_link	?

mdl_customfield_field

- id
- shortname

mdl_customfield_data

- id
- fieldid (FK zu mdl_customfield_field)
- instanceid (FK courseid)
- value

Course

Anzeige	Short name	fieldid
GitHub Classroom	classroom_name	1

mdl_customfield_field

- id
- shortname

mdl_customfield_data

- id
- fieldid (FK zu mdl_customfield_field)
- instanceid (FK courseid)
- value

User

Anzeige	Short name	fieldid
GitHub Benutzername	github_username	1

mdl_user_info_field

- id
- shortname

mdl_user_info_data

- userid
- fieldid (FK auf id in mdl_user_info_field)
- data

mdl_assignfeedback_comments

- id
- assignment (FK auf id in mdl_assign)
- grade
- commenttext
- commentformat

Use Cases

Bewertung aus Repo übernehmen (Umgesetzt)

Lernender macht einen Push ins Repository. Die GitHub Actions bewerten die Abgabe und vergeben eine Punktzahl.

1. Workflow-Step in GitHub Actions
2. FGIT API
3. Pythonscript via CRON

Classroom anlegen

Im Moodle-Kurs wird der Classroom-Name (owner/name) eingetragen.

1. Prüfen ob der Classroom existiert
2. Falls nein: Anlegen des Classrooms
3. Loop über Teilnehmer
 1. In Classroom als Student eintragen (falls noch nicht existiert).

Probleme

- Kein API / CLI von GitHub verfügbar
- Eintragen der Teilnehmer bei Änderungen an den Teilnehmern, z.B. neuer Lernender

Assignment im Classroom erstellen

In der Moodle-Aktivität (z.B. Aufgabe) wird ein Template-Repo eingetragen.

1. Lesen des Classrooms aus Kurs
2. Prüfen ob ein Assignment mit diesem Namen existiert
3. Falls nein:
 1. Assignment anlegen
 2. Link zum Assignment in Moodle-Aktivität übertragen

Probleme

- Kein API / CLI von GitHub vorhanden

From:

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

Permanent link:

<https://wiki.bzz.ch/howto/git/grading/autograding2>

Last update: **2024/03/28 14:07**

