

LU13c - GitHub Workflow

Quelle: <https://docs.github.com/en/actions/using-workflows>



Ein Workflow ist ein automatisierter Prozess, der einen oder mehrere Jobs ausführt.

GitHub Workflows werden in YAML-Dateien definiert, die im Ordner `.github/workflows` unseres Repositories gespeichert werden. Jeder Workflow benötigt:

- Einen oder mehrere Events, welche den Workflow auslösen.
- Einen oder mehrere Jobs, die in einer virtuellen Umgebung (*runner machine*) ausgeführt werden.
- Jeder Job besteht aus einem oder mehreren Schritten.

Beispiel: Copy Issues

Wenn du im GitHub Classroom ein Assignment akzeptierst, wird eine persönliche Kopie des Vorlage-Repositories erstellt. Dabei werden aber allfällige Issues in der Vorlage nicht kopiert. Dieser Workflow und das zugehörige Python Skript ermöglicht es, die Issues in die persönlichen Repositories zu kopieren.

```
name: GitHub Classroom Workflow

on: push

env:
  DEVOPS_DIR: devops
  GHSECRET: ${{ secrets.GITHUB_TOKEN }}

permissions:
  checks: write
  actions: read
  contents: read

jobs:
  copy-issues:
    # copy all issues from the source repo to the students repo
    env:
      SCRIPT_REPO: BZZ-Commons/copy-issues
      SOURCE_REPO: BZZ-Commons/copy-issues # TODO set the owner/name of the
      source repo
      TARGET_REPO: ${{ github.repository }}
      ADD_LABELS: "true" # should the labels for the issues be added
    if: ${{ contains(github.actor, 'classroom') }}
```

```

name: CopyIssues
runs-on: ubuntu-latest
steps:
  - name: Install dependencies
    run: |
      python -m pip install --upgrade pip
      pip install pyGithub
      if [ -f requirements.txt ]; then pip install -r requirements.txt;
    fi
  - name: Check-out the IssueCopy Repo
    uses: actions/checkout@v2
    with:
      repository: ${env.SCRIPT_REPO}
      path: ${env.DEVOPS_DIR}
  - name: Copy the Issues
    run: python ${env.DEVOPS_DIR}/issues.py
    shell: sh

```

Erklärungen

name: GitHub Classroom Workflow

Dieser Name erscheint im Tab „Actions“ um den Workflow zu identifizieren.

on: push

Der Event um diesen Workflow auszulösen; er wird bei jedem push ins Repository ausgelöst. Für eine manuelle Auslösung würde ich `workflow_dispatch` verwenden.

jobs:
copy-issues:

Ab hier werden die Verarbeitungen definiert. In diesem Beispiel gibt es nur einen Job.

```

env:
  SCRIPT_REPO: BZZ-Commons/copy-issues
  SOURCE_REPO: BZZ-Commons/copy-issues # TODO set the owner/name of the
  source repo
  TARGET_REPO: ${github.repository}
  ADD_LABELS: "true"

```

Hier werden einige Umgebungsvariablen definiert. Auf diese Werte kann das Pythonskript zugreifen.

```
if: ${contains(github.actor, 'classroom')}
```

Nur wenn der Push durch den `classroom` Bot erfolgt ist, wird der Job durchgeführt. Dadurch werden die Issues nicht bei jedem Push neu kopiert und überschrieben.

```
runs-on: ubuntu-latest
```

Hier wird die Umgebung (*runner machine*) für den Job festgelegt.

```
steps:
  - name: Install dependencies
    run: |
      python -m pip install --upgrade pip
      pip install pyGithub
      if [ -f requirements.txt ]; then pip install -r requirements.txt;
    fi
  - name: Check-out the IssueCopy Repo
    uses: actions/checkout@v2
    with:
      repository: ${env.SCRIPT_REPO}
      path: ${env.DEVOPS_DIR}
  - name: Copy the Issues
    run: python ${env.DEVOPS_DIR}/issues.py
    shell: sh
```

Nun folgen die Steps mit der eigentlichen Verarbeitung:

1. Installiere alle Abhängigkeiten für das Python Skript.
2. Lies (*check-out*) das Repository mit den Issues.
3. Führe das Python Skript `issues.py` aus.

Falls du das Python Skript anschauen möchtest, du findest es unter
<https://github.com/BZZ-Commons/copy-issues/blob/main/issues.py>.

[M450-LU13](#)



Marcel Suter

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

Permanent link:
<https://wiki.bzz.ch/modul/m450/learningunits/lu13/workflow>

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

