

CI - Anleitung GitHub Actions

| | |
|---|--|
| 1 | Login www.GitHub.com |
| 2 | Template verwenden → Use this Template (Button) https://github.com/mgrum/flask-example-cicd-template |
| 3 | Branch erstellen (Wichtig: wir wollen nicht auf main entwickeln) z.B. develop-workflows |

Workflow **DEV - Build**

| | |
|---|--|
| 4 | Workflow erstellen mit: A. Verzeichnis und Datei .github/workflows/dev_build.yml anlegen B. https://github.com/actions/starter-workflows/blob/618434daabf54b2f994b19c9e608c46db412c6a5/ci/python-app.yml kann als Vorlage für den Inhalt genutzt werden |
| 5 | Name des Workflows anpassen: name: DEV - Build and Unittest |
| 6 | Bedingung anpassen: Wir wollen dass dieser Dev Workflow für alle Branches außer main läuft ... on: push: branches-ignore: [main] |
| 7 | Code ergänzen: Dieser Befehl wird benötigt, da die Applikation das Package "Cython" nutzt und sonst nicht funktionieren würde. ... - name: Install dependencies run: python -m pip install --upgrade pip pip install flake8 pytest |

| | |
|---|---|
| | <pre> if [-f requirements.txt]; then pip install -r requirements.txt; fi - name: Install project run: pip install -e Hilfestellung 1: https://pastebin.com/iiGuhr6i </pre> |
| 8 | <p>Jobs anpassen: Wir wollen Build und Test in einzelne Jobs trennen Dazu muss pytest aus dem Job build raus und ein Job test angelegt werden</p> <pre> ... test: needs: [build] runs-on: ubuntu-latest steps: - uses: actions/checkout@v2 - name: Set up Python 3.x uses: actions/setup-python@v2 with: python-version: '3.x' - name: Display Python version run: python -c "import sys; print(sys.version)" - name: Install dependencies run: python -m pip install --upgrade pip pip install pytest pip install -r requirements.txt - name: Install project run: pip install -e . - name: Test with pytest run: pytest </pre> |

| | |
|----|--|
| 9 | Workflow Datei speichern / Committen |
| 10 | Auf Actions Tab drücken und der Workflow sollte nun starten |
| 11 | <p>Gelaufenen Workflow öffnen:</p> <p>A. Build sollte erfolgreich sein</p> <p>B. Test sollte einen Fehler werden → 1/8 Unit-Tests müsste fehlgeschlagen sein</p> <p>In der Ausgabe des Workflows sollte man den Fehler finden können: flaskr/app.py → Zeile 16 → <code>return {"hello": "world"}</code></p> |
| 12 | <p>Fehler beheben:</p> <p><code>return {"hello": "world"}</code> → <code>return {"hello": "IWS"}</code></p> <p>Datei Speichern / Committen</p> |
| 13 | Auf Actions Tab drücken und Workflow sollte nun erfolgreich funktionieren :-) |

Workflow **STAGE** - Test

| | |
|----|--|
| 14 | <p>Einen neuen Workflow anlegen:</p> <p>.github/workflows/stage_test.yml</p> <p>Ihr könnt den Code von dev_build.yml kopieren</p> |
| 15 | <p>Name und Bedingung anpassen:</p> <p>name: STAGE - Matrix Test</p> <p>on:</p> <p> pull_request:</p> <p> branches:</p> <p> - main</p> <p> ...</p> |
| 16 | <p>Test anpassen:</p> <p>Nun soll nicht nur schnell getestet werden ob es überhaupt funktioniert, sondern ein intensiver Test auf mehreren OS und Python Versionen laufen</p> |

| | |
|----|---|
| | <pre> ... test: needs: [build] runs-on: ubuntu-latest strategy: matrix: os: [ubuntu-latest, macos-latest, windows-latest] python-version: [3.6, 3.9] steps: - uses: actions/checkout@v2 - name: Set up Python \${ matrix.python-version } uses: actions/setup-python@v2 with: python-version: \${ matrix.python-version } ... </pre> <p>Hilfestellung 2: https://pastebin.com/LXCSueL2 Speichern / Committed nicht vergessen</p> |
| 17 | <p>Pull Request stellen Nun einen PR von develop-workflows gegen main stellen</p> |
| 18 | <p>Innerhalb des PR kann man nun die Workflows DEV und STAGE beobachten Alternativ: Auf Actions Tab drücken</p> |
| 19 | <p>Wenn alle checks erfolgreich waren, könnte der PR gemerged und der branch develop-workflows gelöscht werden.</p> |

Einzelne Lösungen der Workflows:

dev_build.yml: <https://pastebin.com/YZBKlk1i>

stage_test.yml: <https://pastebin.com/hbchPmbw>

Nun sollten wir auf dem Stand von der **Musterlösung** sein:
<https://github.com/mgrum/flask-example-cicd-solution>

Nützliche Links:

[GitHub Actions - Workflow Syntax](#)

[Github Actions - Supported Software](#)