

Spickzettel: GitHub Actions – Was sind Actions?

1. Was sind GitHub Actions?

- Automatisierung von Workflows direkt in GitHub.
- Erlaubt **CI/CD (Continuous Integration & Deployment)**.
- Nutzt **Workflows**, die in `.github/workflows/` definiert sind.
- Kann Tests, Builds, Deployments & andere Automationen ausführen.

2. Grundaufbau einer Action

- Eine Action besteht aus **Jobs**, die in **Schritten (Steps)** definiert sind.
- Wird als **YAML-Datei** gespeichert.

Beispiel einer einfachen GitHub Action (`ci.yml`)

```
name: CI Workflow
on: [push, pull_request]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Repository auschecken
        uses: actions/checkout@v3
      - name: Node.js einrichten
        uses: actions/setup-node@v3
        with:
          node-version: '16'
      - name: Abhängigkeiten installieren
        run: npm install
      - name: Tests ausführen
        run: npm test
```

3. Wichtige Konzepte

- **Events:** Bestimmen, wann ein Workflow startet (z. B. `push`, `pull_request`, `schedule`).
- **Jobs:** Enthalten eine Reihe von Schritten (`steps`), die in einer Umgebung ausgeführt werden.
- **Steps:** Einzelne Befehle oder vorgefertigte Aktionen (`uses:`).
- **Runner:** Virtuelle Maschinen, auf denen die Jobs ausgeführt werden (`ubuntu-latest`, `windows-latest`, `macos-latest`).

4. Arten von GitHub Actions

- **Offizielle GitHub Actions** (`actions/checkout`, `actions/setup-node`)
- **Community Actions** (über GitHub Marketplace verfügbar)
- **Eigene Custom Actions** (als eigene `.yml`-Dateien oder Docker-Container)

5. Wichtige Aktionstypen

- **uses** → Eine bestehende Action ausführen:
 - - name: Repository auschecken
 - uses: actions/checkout@v3
- **run** → Eigene Shell-Befehle ausführen:
 - - name: Test ausführen
 - run: npm test
- **with** → Parameter an eine Action übergeben:
 - - name: Node.js einrichten
 - uses: actions/setup-node@v3
 - with:
 - node-version: '16'
- **env** → Umgebungsvariablen setzen:
 - - name: Setze Umgebungsvariable
 - run: echo "API_KEY=\$API_KEY"
 - env:
 - API_KEY: \${ secrets.API_KEY }
- **if** → Bedingte Ausführung:
 - - name: Nur auf `main` ausführen
 - run: echo "Deployment läuft"
 - if: github.ref == 'refs/heads/main'

6. Häufig genutzte vordefinierte Actions

Action	Funktion
actions/checkout	Klont das Repository in den Runner
actions/setup-node	Installiert Node.js
actions/setup-python	Installiert Python
actions/cache	Speichert Abhängigkeiten für schnellere Builds
actions/upload-artifact	Speichert Dateien aus dem Workflow
actions/download-artifact	Lädt zuvor gespeicherte Artefakte herunter
github/codeql-action	Automatische Sicherheitsanalyse

Best Practices

- Actions in `.github/workflows/` klar strukturieren.
- **Caching nutzen** (actions/cache), um Builds zu beschleunigen.
- **Secrets & Umgebungsvariablen sicher verwalten** (secrets.GITHUB_TOKEN).
- **Jobs parallel ausführen**, wenn möglich (strategy.matrix).