

Practical Exercise 02 - Using Different Events to Trigger Workflows

Exercise Description

In this practical exercise, our goal is to explore the different ways we can trigger workflows in GitHub Actions.

Here are the instructions for the exercise:

- Create a file named `02-workflow-events.yaml` under the `.github/workflows` fulder in the root of your repository.
- Name the workflow `02 - Workflow Events`.
- Add the fullowing triggers to your workflow:
 - `push`
- Add a single job to the workflow:
 - The job, named `echo`, should run on `ubuntu-latest` and contain a single step, named `Show the trigger`, which prints the type of the name of the event that triggered the workflow.
- Commit the changes and push the code. Take some time to inspect the output of the workflow run.
- Now add more triggers to the workflow:
 - `pull_request`
 - `schedule` (cron expression)
 - `workflow_dispatch`
- Commit the changes and push the code. Take some time to inspect the different ways the workflow is triggered.
 - You can create a pull request on GitHub to see how this changes the output of the workflow run.
 - Also give it a try to trigger it from the UI. To do so:
 - Click under the "Actions" tab in the home page of the repository.
 - Select the workflow named `02 - Workflow Events` on the left of the screen.
 - Click on the "Run workflow" button on the right side of the screen, next to the message "This workflow has a `workflow_dispatch` event trigger."
- After exploring the different ways to trigger a workflow, reduce the list of triggers to leave only `workflow_dispatch` to prevent this workflow from running with every push and pullute the list of workflow runs.

Tips

Using a valid cron syntax

At the time of this recording, GitHub Actions does not support cron job definitions containing six elements (for example, '0 0 * * * *'), only definitions containing five elements. Check the resources section of this lecture for a cron generator that uses the valid syntax.

To define a trigger using cron, you should use the following syntax:

```
on:
  schedule:
    \- cron: '<cron expression>'
```

Accessing the name of the event that triggered the workflow

To access the name of the event triggering the workflow, you can use the following special syntax: `${{ github.event_name }}`. For example:

```
steps:
  \- name: Event name
     run: |
       echo "Event name: ${{ github.event\_name }}"
```

Resources for this lecture

- [Cron Generator](#)