GitHub Actions Workflow Triggers

GitHub Actions workflows are triggered by specific events that occur within your repository. These triggers define when a workflow should run, automating various tasks like continuous integration, testing, and deployment.

Here are some common triggers for GitHub Actions workflows:

- **Push Events**
- **Pull Request Events**
- **Scheduled Events**
- **Manual Triggers**

Testing Event Triggers Workflow

Push Events

Trigger a workflow when code is pushed to a specific branch.

Example YAML:

```yaml
on:
  push:
    branches:
      - main
      - release/*
```

Pull Request Events

Trigger a workflow when pull requests are opened, updated, or synchronized.

Example YAML:

```yaml
on:
  pull_request:
    branches:
      - main
```

Scheduled Events

Schedule workflows to run at specific times or intervals using cron expressions.

Example YAML:

```
on:
  schedule:
    - cron: '*/5 0 * * *' # Run every 5 min
```

Manual Triggers

Allow manual triggering of workflows through the GitHub Actions UI or API.

Example YAML:

```
on:
  workflow_dispatch:
```

These examples demonstrate various ways to trigger GitHub Actions workflows based on specific events and conditions. You can customize them to suit your project's needs and automate your development processes effectively.

Testing Event Triggers Workflow

1. Create a Workflow File:

- Create a new file named `04-01-Workflow-Events.yml` in the `.github/workflows` directory of your repository.

  ```
  Name: 04-01-Workflow Events
  Description: Demonstrates different GitHub Actions trigger events.

  on:
    push: # Trigger the workflow on a push event
    # workflow_dispatch: # Uncomment to trigger the workflow manually via the
  GitHub UI
    # pull_request: # Uncomment to trigger the workflow on pull requests
    # schedule: # Uncomment to trigger the workflow on a schedule
    #   - cron: '*/5 0 * * *' # Uncomment to run the workflow every 5 minutes

  jobs:
    echo:
      runs-on: ubuntu-latest
      steps:
        - name: Show the trigger
          run: echo "I've been triggered by a(n) ${{ github.event_name }}
  event."
  ```

- Copy and paste the provided YAML configuration into this file.

2. Understanding the Workflow:

- The workflow is triggered by a `push` event, meaning it will run whenever code is pushed to the repository.
- Optionally, you can uncomment and use other trigger events such as `workflow_dispatch`, `pull_request`, or `schedule` by removing the # symbol and specifying additional configurations if needed.
- The workflow defines a single job named echo that runs on an Ubuntu environment (runs-on: ubuntu-latest).
- The job consists of a single step:

3. Show the Trigger:

- This step runs a shell command to echo the event that triggered the workflow.
- It uses the `${{ github.event_name }}` syntax to dynamically display the name of the trigger event.

4. Testing the Workflow:

- Commit and push the workflow file (`04-01-Workflow-Events.yml`) to your repository.
- Make sure you have code changes ready to push to trigger the workflow (since it's triggered by a `push` event).
- Go to the "Actions" tab in your GitHub repository to view the workflow runs.
- Monitor the workflow execution and check the output of the step to ensure it displays the correct trigger event.

6. Optional: Triggering Manually or via Pull Requests:

- If you uncomment and use the `workflow_dispatch` or `pull_request` events, you can trigger the workflow manually via the GitHub UI or when pull requests are opened or synchronized.
- For the `schedule` event, you can specify a cron schedule to run the workflow at specific times.