Using Event Filters and Activity Types

GitHub Actions workflows are highly customizable and can be triggered by various events and activities.

GitHub Actions allows you to automate your workflows based on specific events and activities. By defining event filters and activity types in your workflow configuration, you can precisely control when your actions run.

Event Filters

Using on

You can specify when a workflow should run by using the `on` keyword in your workflow YAML file. This is the primary way to define the event that triggers your workflow.

```
on:
  push:
    branches:
      - main
```

In this example, the workflow is triggered on a push event to the main branch.

Filtering by Event Types

GitHub Actions supports a wide range of event types, such as push, pull_request, release, and more. You can filter your workflow by these event types.

```
on:
  push:
    branches:
      - main
```

Filtering by Branches

You can further narrow down your workflow's trigger by specifying the branches on which the event should occur.

```
on:
  push:
    branches:
      - main
      - feature/*
```

Here, the workflow runs on a push event to either the main branch or branches starting with "feature/."

Filtering by Paths

If you want to trigger your workflow only when specific files or directories are modified, you can filter by paths.

```
on:
  push:
    paths:
      - 'src/**'
      - 'docs/'
```

This workflow is triggered on a push event that modifies files under the src/ directory or any file or folder in the docs/ directory.

Filtering by Tags

You can also use tags to trigger your workflow when specific tags are created.

```
on:
  push:
    tags:
      - v*
```

In this example, the workflow runs when a tag starting with "v" is pushed.

Activity Types

GitHub Actions provides several activity types per event. The best place to find a comprehensive list of all available activity types is the GitHub Actions documentation.

- **Pull Request Activity Types**

Pull Request Activity Types

Pull requests offer a variety of activity types. The example below filters the activity types down to `opened` and `synchronize`. This means that the workflow will trigger when a pull request is opened, or when a new commit is pushed to the `head` of the PR branch:

```
on:
  pull_request:
    types:
      - opened
      - synchronize
```

Testing Event Filters and Activity Types

1. Create Workflow Files:

- Create two new files named `07-01-filters-activity-types.yml` and `07-02-filters-activity-types.yml` in the `.github/workflows` directory of your repository.
- **07-01-filters-activity-types.yml**

```
name: 07-01-Event Filters and Activity Types

on:
  pull_request:
    types:
      - opened
      - synchronize
    branches:
      - main

jobs:
  echo:
    runs-on: ubuntu-latest
    steps:
      - run: echo "Running whenever a PR is opened or synchronized AND base branch is main"
```

- **07-02-filters-activity-types.yml**

```
name: 07-02-Event Filters and Activity Types

on:
  pull_request:
    types:
      - closed
    branches:
      - main

jobs:
```

```yaml
  echo:
    runs-on: ubuntu-latest
    steps:
      - run: echo "Running whenever a PR is closed"
```

- Copy and paste the respective YAML configurations into each file.

2. Understanding the Workflows:

- 07-01-filters-activity-types.yml:
  - This workflow runs whenever a pull request is opened or synchronized (`types: opened, synchronize`) and the base branch is `main`.
  - It consists of a single job named `echo`, which runs on the `ubuntu-latest` environment.
  - The job includes a single step that echoes a message indicating that the workflow is running whenever a PR is opened or synchronized and the base branch is `main`.
- 07-02-filters-activity-types.yml:
  - This workflow runs whenever a pull request is closed (`types: closed`) and the base branch is `main`.
  - Similar to the first workflow, it also contains a single job named `echo` that runs on the `ubuntu-latest` environment.
  - The job includes a single step that echoes a message indicating that the workflow is running whenever a PR is closed.

3. Testing the Workflows:

- Commit and push the workflow files (`07-01-filters-activity-types.yml` and `07-02-filters-activity-types.yml`) to your repository.
- Create or open a pull request on the `main` branch of your repository.
- For the first workflow, ensure that the message specified in the step is echoed when the pull request is opened or synchronized.
- After merging or closing the pull request, verify that the message specified in the second workflow is echoed.

4. Reviewing Workflow Runs:

- Go to the "Actions" tab in your GitHub repository to view the workflow runs.
- Check the status and output of each workflow run to confirm that they executed as expected.
- If any issues arise, review the workflow configurations and troubleshoot any errors.