

Lesson 1 Demo 2

GitHub action to understand Customization of Actions

Action: Resuable unit of code that perform specific tasks within workflows . We will understand how the developers can customize and extend the actions to fit their specific requirments and the benefits this customization brings to the SDLC

Objective: You were looking for implementing GitHub Actions workflow for customizing actions

Tools required: GitHub Actions

Prerequisites: None

Steps to be followed:

1. Login to GitHub.com.
2. Create a new workflow file to perform basic execution of command.

Step 1: Login to GitHub.com

- 1.1 Sign in to the GitHub portal at <https://github.com>
- 1.2 Navigate to your repository main page where we are going to create a workflow file.
- 1.3 In case the repository is not available fork below mentioned repository in your GitHub account.
<https://github.com/Simplilearn-Edu/MavenBuild>

Step 2: Create a new workflow file to perform basic execution of command.


- 2.1 Once you are on GitHub main repository page, we can get started with creating a new Workflow file.
- 2.2 Create a .github/workflows directory in your repository in case directory does not exists.
- 2.3 In workflow directory, create a new workflow file with name custom_action.yml.
- 2.4 Add below workflow content in above created file.

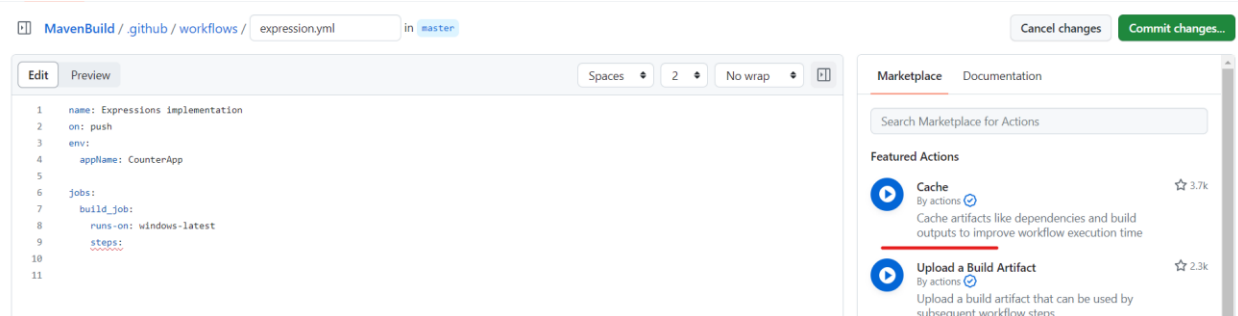
```

name: Expressions implementation
on: push
env:
  appName: CounterApp

jobs:
  build_job:
    runs-on: windows-latest
    steps:

```

- 2.5 To the right of the editor, use the GitHub Marketplace sidebar to browse actions. Actions with the  badge indicate GitHub has verified the creator of the action as a partner organization.



- 2.6 Select the desired action and select to get the code snippet and place them under steps section in workflow. Here, we are using the below code:

```

- name: Cache
  uses: actions/cache@v1.2.1

```

with:

A directory to store and save the cache

path: ""

An explicit key for restoring and saving the cache

key: ""

An ordered list of keys to use for restoring the cache if no cache hit occurred for key

restore-keys: "" # optional

The screenshot shows the GitHub Actions workflow editor interface. On the left, the workflow file 'expression.yml' is open, showing a job named 'build_job' with a step named 'steps:'. On the right, the 'Cache' action is selected from the Marketplace. The 'Installation' section shows the following snippet to be pasted into the workflow file:

```

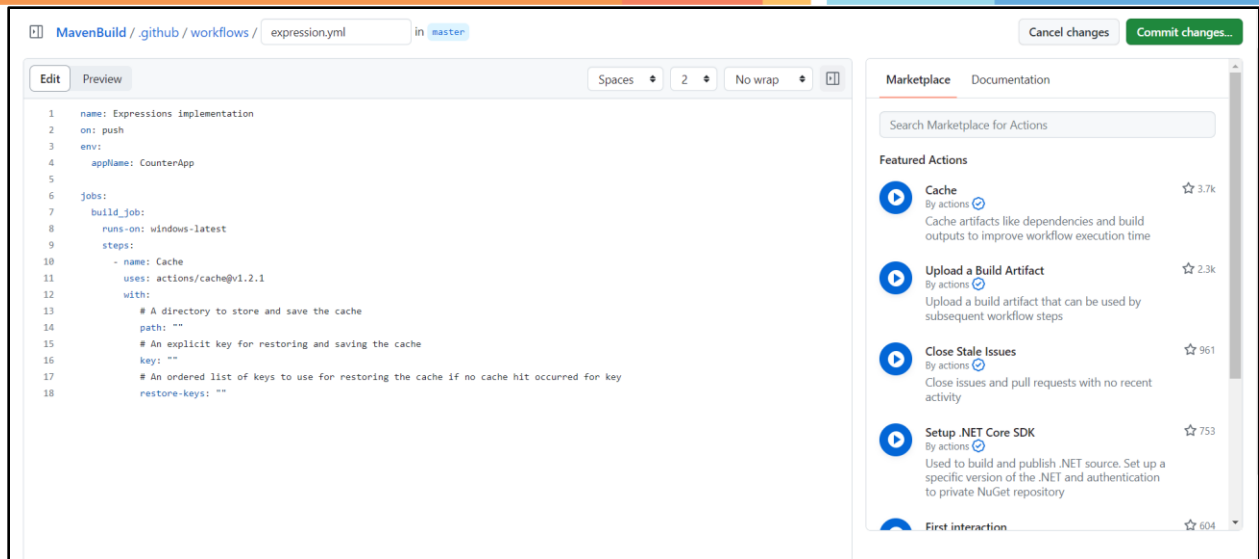
- name: Cache
  uses: actions/cache@v1.2.1
  with:
    # A directory to store and save the cache
    path: ""
    # An explicit key for restoring and saving the cache
    key: ""
    # An ordered list of keys to use for restoring the cache
    restore-keys: # optional
  
```

The screenshot shows the GitHub Actions workflow editor interface. On the left, the workflow file 'expression.yml' is open, showing a job named 'build_job' with a step named 'steps:'. On the right, the 'Cache' action is selected from the Marketplace. The 'Installation' section shows the following snippet to be pasted into the workflow file:

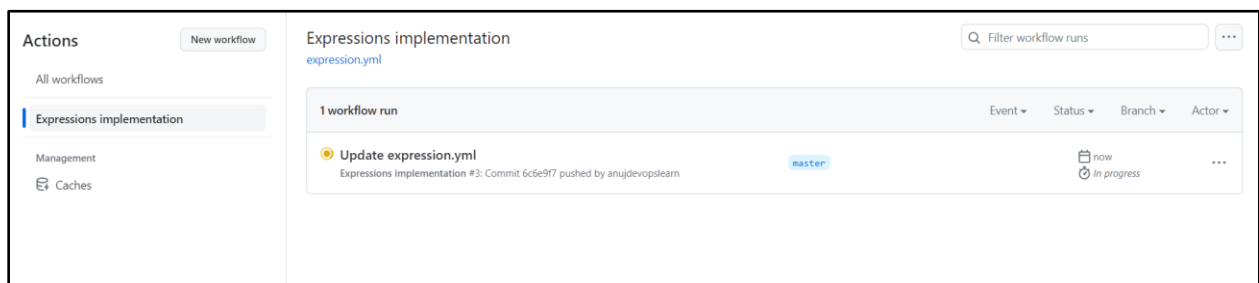
```

- name: Cache
  uses: actions/cache@v1.2.1
  with:
    # A directory to store and save the cache
    path: ""
    # An explicit key for restoring and saving the cache
    key: ""
    # An ordered list of keys to use for restoring the cache
    restore-keys: # optional
  
```

2.7 Once file is created, click on Commit new file to save workflow file in your code repository.



2.8 Next navigate to Actions tab on your repository to access workflow execution page as below:



2.9 From left sidebar you can select workflow which you want to access execution logs.

2.10 Select latest workflow execution run to access execution logs for the jobs configured in workflow.