

Working with Github and Git – Activity 1



accenture

Index

Exercise: Working with Github and Git	3
---	---

Version	Revision Date	Description	Author	Reviewed by	Approved by
V1.0	21-Apr-23	Initial Version	Rakhi Parashar	Rathna Perumalsamy	CPCL, Sreenivasa Rao

Exercise: Working with Github and Git

Prerequisite:

1. Working knowledge of Git and SCM

Walkthrough:

1. Install Git
2. Create project in Github server
3. Import Project and observe webhooks is created.
4. Running Git commands using gitbash.

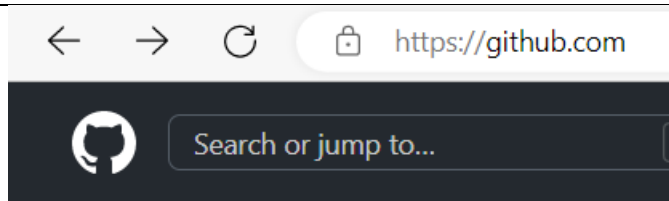
Steps:

1. Install Git

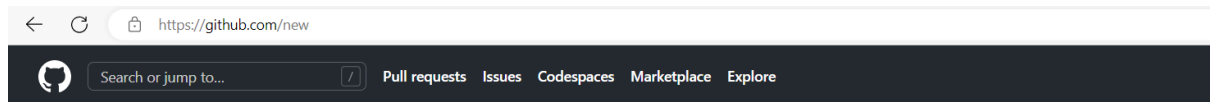
1	<p>Following are the steps to install Git:</p> <ol style="list-style-type: none">a. Click on Git Download for Windows from the below URLb. https://git-scm.com/downloadsc. Once download is complete, go to downloads folder and go for right click Run as administrator option git.exe software.d. Select Accenture Business permissions.e. Keep remaining options as default and click Install button.f. If Git installation is completed successfully, then follow these steps.g. Navigate and select Git Bash
---	---

2. Import project on Github server

1	<p>Signin to the Github server</p> <ol style="list-style-type: none">a. You are required to signup to Github at https://github.com/b. While signup, please provide your mail id as an email address.c. (Confirmation link will be sent).d. Once the account is created, login to the account.
2	<p>Click on new project</p>



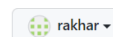
Top Repositories



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *



Repository name *



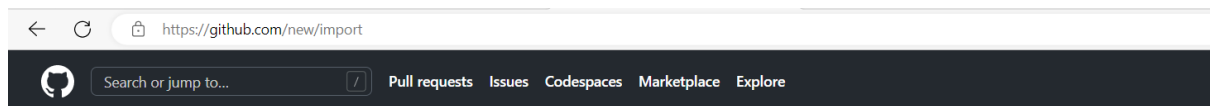
Great repository names are short and memorable. Need inspiration? How about [super-duper-winner?](#)

Description (optional)



Provide the below URL and click on Begin Import.

`https://github.com/rakhar/webhook-test.git`



Import your project to GitHub

Import all the files, including revision history, from another version control system.

Support for importing Mercurial, Subversion and Team Foundation Version Control (TFVC) repositories will end on October 17, 2023. For more details, see the [changelog](#).

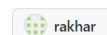
Your old repository's clone URL

`https://svn.example.org/code/svn`

Learn more about the types of [supported VCS](#).

Your new repository details

Owner *



Repository name *

`https://github.com/rakhar`

⚠ Your new repository will be created as `https-github.com-rakhar-webhook-test`.

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

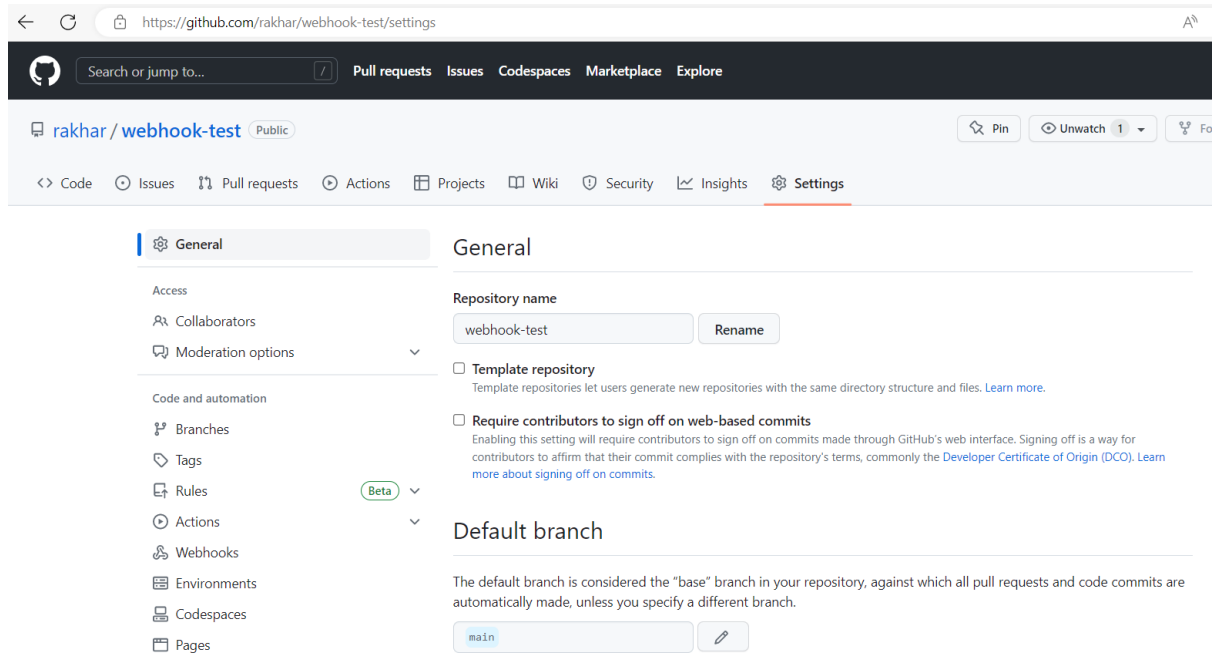
You choose who can see and commit to this repository.

③ You are creating a public repository in your personal account.

[Cancel](#)

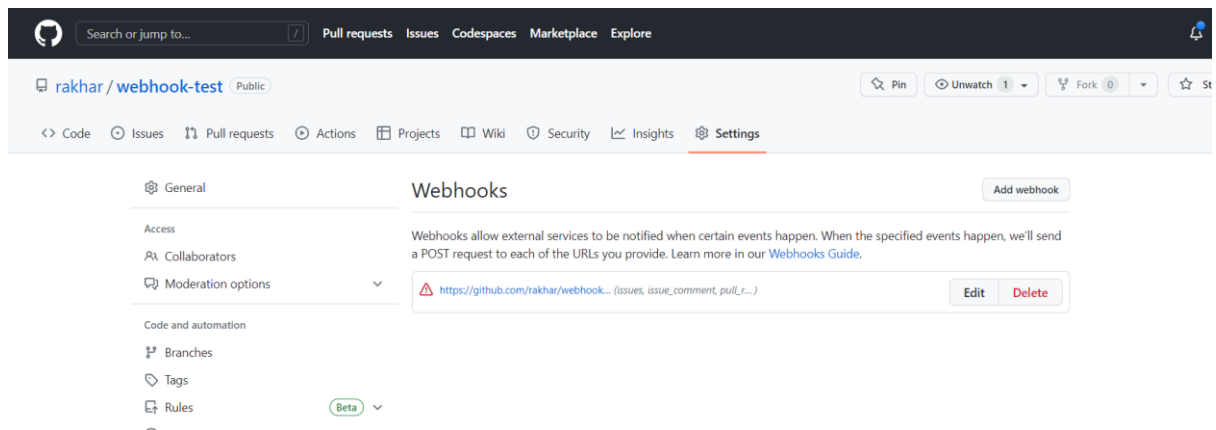
[Begin import](#)

3 Goto settings:



The screenshot shows the GitHub repository settings page for 'rakhar/webhook-test'. The 'General' tab is selected in the left sidebar. The main content area shows the 'General' settings, including the repository name 'webhook-test' with a 'Rename' button. Below this are checkboxes for 'Template repository' and 'Require contributors to sign off on web-based commits'. The 'Default branch' is set to 'main'. The left sidebar lists various settings categories: Access (Collaborators, Moderation options), Code and automation (Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, Pages). The 'Webhooks' category is highlighted with a green circle.

Click on Webhooks and then click on Add webhook:



The screenshot shows the GitHub repository settings page for 'rakhar/webhook-test', specifically the 'Webhooks' tab. The left sidebar shows the 'Webhooks' category highlighted with a green circle. The main content area shows the 'Webhooks' settings, including a description of webhooks and a list of existing webhooks. One webhook is listed with the URL 'https://github.com/rakhar/webhook... (issues, issue_comment, pull_r...)'. There are 'Edit' and 'Delete' buttons for this webhook. An 'Add webhook' button is visible in the top right corner.

☒ **Issues**

Issue opened, edited, deleted, transferred, pinned, unpinned, closed, reopened, assigned, unassigned, labeled, unlabeled, milestone, demilestoned, locked, or unlocked.

☐ **Merge groups**

Merge Group checks requested.

☐ **Milestones**

Milestone created, closed, opened, edited, or deleted.

☐ **Page builds**

Pages site built.

☐ **Project columns**

Project column created, updated, moved or deleted.

☐ **Pull request review comments**

Pull request diff comment created, edited, or deleted.

☒ **Pull request reviews**

Pull request review submitted, edited, or dismissed.

☒ **Pushes**

Git push to a repository.

☐ **Labels**

Label created, edited or deleted.

☐ **Meta**

This particular hook is deleted.

☐ **Packages**

GitHub Packages published or updated in a repository.

☐ **Project cards**

Project card created, updated, or deleted.

☐ **Projects**

Project created, updated, or deleted.

☐ **Pull request review threads**

A pull request review thread was resolved or unresolved.

☒ **Pull requests**

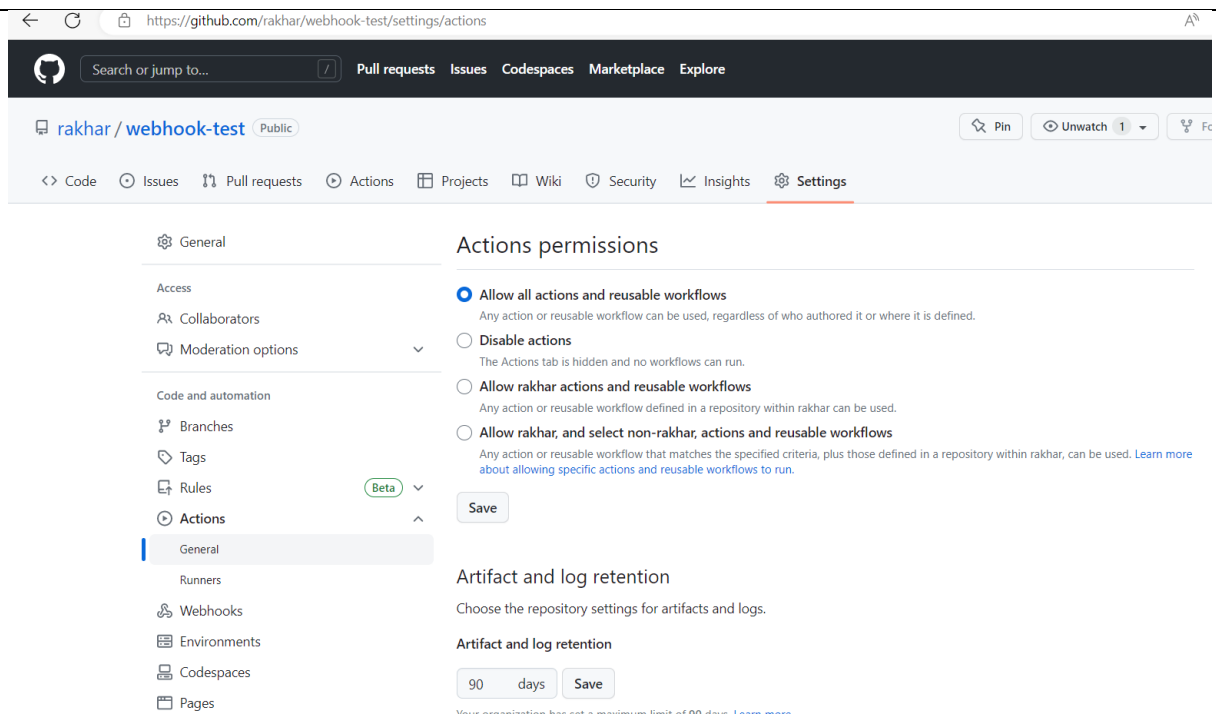
Pull request assigned, auto merge disabled, auto merge enabled, closed, converted to draft, demilestoned, dequeued, edited, enqueued, labeled, locked, milestone, opened, ready for review, reopened, review request removed, review requested, synchronized, unassigned, unlabeled, or unlocked.

☐ **Registry packages**

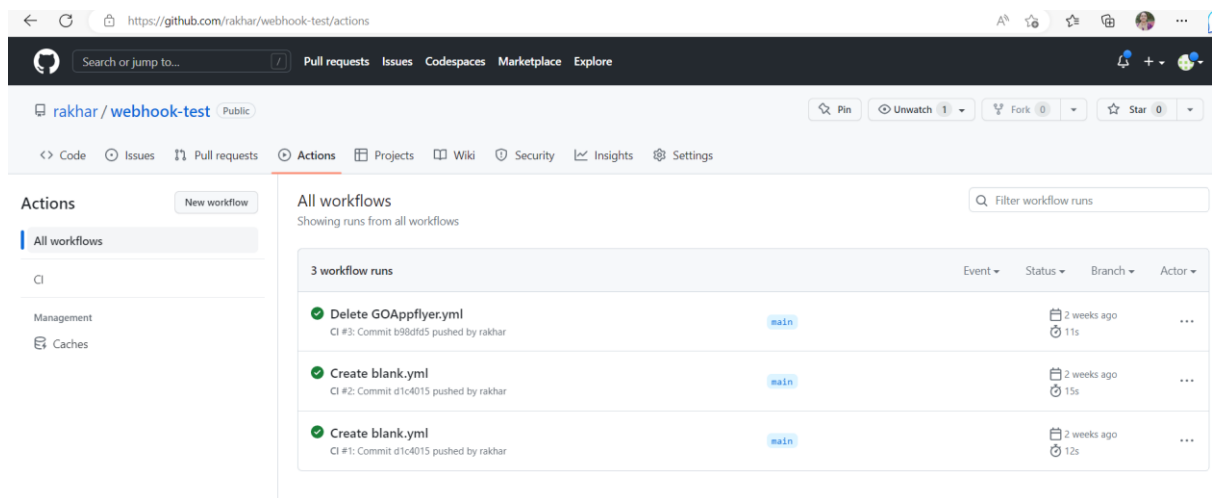
Registry package published or updated in a repository.

Make sure above check boxes are selected and then add webhook.

4 Also in Settings, under Actions make sure you select as below:



Now make any change in Readme or any file and commit. Observe the Github Pipeline triggered automatically.



3. Running Git Commands based:

1	//Updating Config list git config --list git config --global user.name "[name]" git config --global user.email "[email address]"
2	// View total no. of branches git branch

	<pre>// to create a develop branch git branch develop</pre>
3	<pre>//Checkout branch or switching to a branch or //moving to any new branch. git checkout develop</pre>
4	<pre>//Adding new files: touch filename //How to make changes using vi vi filename press i type ur lines of code press escape key press :wq enter key</pre>
5.	<pre>//To see the status of the tracked files in staging area git status</pre>
6.	<pre>//git add command will add any modifications of source code or any new additions to staging area. git add filename git add . //To untrack the modifications done and added to staging you can use restore or reset. i.e to revert back. git restore --staged testfile.txt git add . git commit -m "comments" git log git reset --hard commitID</pre>
7.	<pre>//To see logs for the commits done so far. git log git show commitID</pre>
8.	<pre>//Will show modifications done in source code Git diff git diff --staged</pre>
9.	<pre>//Will pull the modifications done by anyone in same project to your local Git pull //Merging changes and pushing to remote git merge branchname git push</pre>
10.	<pre>//Stash command helps you to save or park your changes for temporary purpose and continue making further changes, whenever we are ready we can bring back the saved/parked changes and do the commit and push to remote repository. It follows LIFO. git stash save git stash list git stash drop git stash pop git stash list git stash pop stash@{1}</pre>

	git stash pop stash@{0}
11.	//Use tags for your multiple releases git tag -a <tag-name> -m "tagging message" (creates an annotated tag) git push origin tagname git tag git show tagname
12.	git revert
	//rm - Remove file from git project repository, clean - Remove untracked files or directory from your git //repository git rm filename git clean -n git clean -fd