# Freestyle Project

## Contents

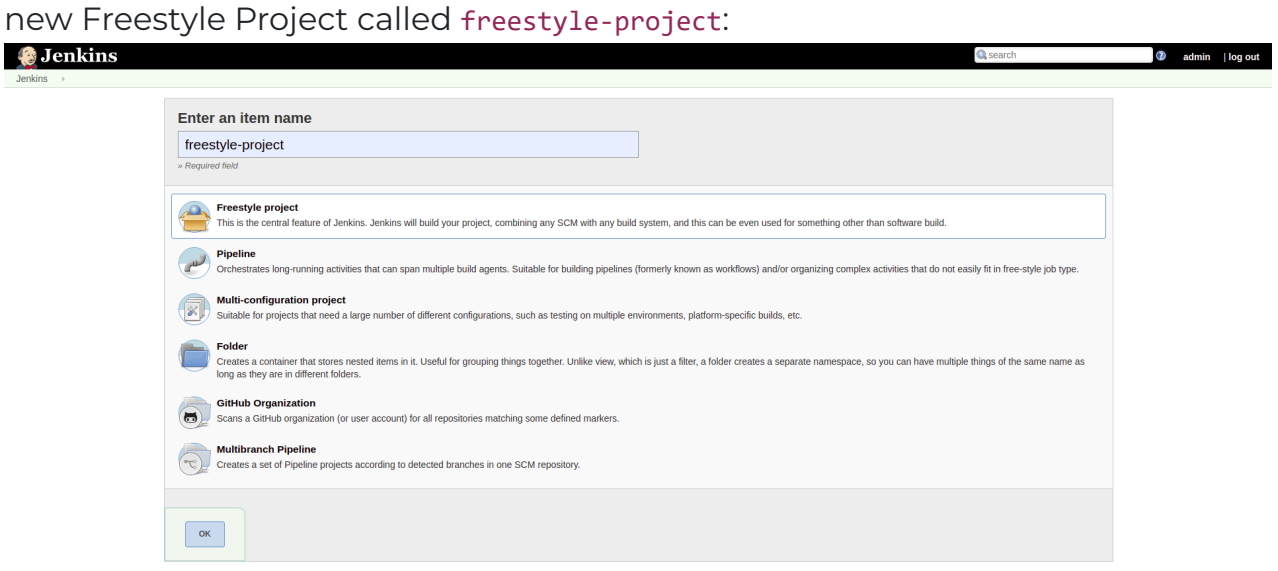


## Overview

Freestyle projects in Jenkins are a type of job you can create for almost any automated task.
These are the best place to start for building any sort of general purpose automation in Jenkins.

## Create a Freestyle Project

Go ahead and select `New Item` from the Jenkins dashboard, and then create a new Freestyle Project called `freestyle-project`:




## Configuration

Once you have created a new job, you will then be redirected to a configuration page for that job.

### Source Code Management

This section is used for configuring a source code repository to download.
The job will download the repository you provide into the jobs workspace.

### Build Triggers

The most simple way to run a Jenkins job is by pressing the build button for the job.
However, jobs can be triggered in many ways - we will usually try to avoid doing this manually.

| Option | Description |
| --- | --- |
| Build Periodically | You can create a schedule here for the job; for example, having it build every hour or at 6:15 PM every Thursday. |
| GitHub Hook | This is where GitHub can send a HTTP POST request; for example, a web hook to your Jenkins server to trigger a build of the job. This must be configured in GitHub and your Jenkins instance must be accessible from the internet for this to work. |
| Poll SCM | This feature can be used if your Jenkins instance is not accessible on the internet. Jenkins will check, using a schedule that you define, whether a change has been made on the configured SCM repository. As soon as a change has been detected, the job will be executed. |

## Build Environment

Various options that can be configured for when the job runs.

| Option | Description |
| --- | --- |
| Delete Workspace Before Build Starts | You will likely want this option checked. The folder where the job runs on the host machine's file system will be deleted before building again. |
| Secret Texts & Files | You may securely use secret texts and files that you have configured in the Credentials section here in the job. These secrets will also be hidden in the Jenkins logs as well. |
| Build | This is likely where you will spend most of your time on a Jenkins job. The most common build step here is `Execute shell`; other options are available, depending on what plugins you have installed. Exactly what your job accomplishes is configured here. |

## Build

The part of the job that is 'executed'.
Functional steps of the job are configured here.

| Option | Description |
| --- | --- |
| Build | Select your build step. This is likely where you will spend most of your time on a Jenkins job. The most common build step here is `Execute shell`; other options are available, depending on what plugins you have installed. Exactly what your job accomplishes is configured here. |

## Post-build Actions

Events to trigger once a build has been completed.

| Option | Description |
| --- | --- |

| Option | Description |
| --- | --- |
| Add Post-build Actions | Select your post-build action. You may want to configure your job to react depending on how the build went. For instance, you could be notified by email if the job failed. You could also publish a report if the job completed successfully. Like with most of the options, the sky is the limit; it all depends on what plugins are installed. |

## Tutorial

These tasks will take you through configuring a very simple Freestyle Project, which will download a script and run it.

### Prerequisites

- GitHub Account
- Jenkins Installed

### Create a New GitHub Repository

GitHub is of course where our code will be, we'll need to prove that Jenkins can download that code and access in the Job.
Setup a repository so we can configure a Jenkins job to access and use it in later steps:

1. Create a *public* GitHub repository for this exercise, you can call it `jenkins-freestyle-project`.
2. Add a script to the repository called `run.sh`, with the contents:

```
echo 'Hello from run.sh!'
```

### Create a Jenkins Job

The Jenkins job is going to be able to:

- download the repository that we created
- run the `run.sh` script

1. Create a new Freestyle Project on Jenkins, you can call this job whatever you like.
2. Configure the Job to download the repository
    - Under *Source Code Management*, select *Git*
    - Enter `https://github.com/[YOUR_USERNAME]/jenkins-freestyle-project`, replacing `[YOUR_USERNAME]` with your GitHub username.
    - If your Git Repository has a main branch, then you will need to change the branch specifier from `*/master` to `*/main`.
3. In the *Build* section, create a build step to *Execute shell* and enter the following:

```
sh run.sh
```

### Run the Job

Now everything is setup, *Save* the changes that were made and the *Build* the job.
Go and check the console output of the build to see that the job has executed, the end of the output should show that the script on the repository has run correctly:

```
+ sh run.sh
+ Hello from run.sh!
+ Finished: SUCCESS
```

### Clean Up

Feel free to now delete the created resources:

- Jenkins job
- GitHub Repository

## Exercises

There are no exercises for this module.