

Developing a third party CLI action

In this article

- Introduction
- Prerequisites
- Example
- Further reading

Learn how to develop an action to set up a CLI on GitHub Actions runners.

Introduction

You can write an action to provide a way for users to access your servers via a configured CLI environment on GitHub Actions runners.

Your action should:

- Make it simple for users to specify the version of the CLI to install
- Support multiple operating systems
- Run in an efficient fashion to minimize run-time and associated costs
- Work across GitHub Enterprise Cloud-hosted and self-hosted runners
- Leverage community tooling when possible

This article will demonstrate how to write an action that retrieves a specific version of your CLI, installs it, adds it to the path, and (optionally) caches it. This type of action (an action that sets up a tool) is often named `setup-$TOOL`.

Prerequisites

You should have an understanding of how to write a custom action. For more information, see "[About custom actions](#)". For a more detailed guide on how to write a custom action, see "[Creating a JavaScript action](#)".

Example

The following script demonstrates how you can get a user-specified version as input, download and extract the specific version of your CLI, then add the CLI to the path.

GitHub provides `actions/toolkit`, which is a set of packages that helps you create actions. This example uses the `actions/core` and `actions/tool-cache` packages.

JavaScript



```
const core = require('@actions/core');
const tc = require('@actions/tool-cache');

async function setup() {
```

```
// Get version of tool to be installed
const version = core.getInput('version');

// Download the specific version of the tool, e.g. as a tarball
const pathToTarball = await tc.downloadTool(getDownloadURL());

// Extract the tarball onto the runner
const pathToCLI = await tc.extractTar(pathToTarball);

// Expose the tool by adding it to the PATH
core.addPath(pathToCLI)
}

module.exports = setup
```

To use this script, replace `getDownloadURL` with a function that downloads your CLI. You will also need to create an actions metadata file (`action.yml`) that accepts a `version` input and that runs this script. For full details about how to create an action, see "[Creating a JavaScript action](#)."

For a full example of how to set up this action, see [example-setup-gh](#).

Further reading

This pattern is employed in several actions. For more examples, see:

- [ruby/setup-ruby](#)
- [google-github-actions/setup-gcloud](#)
- [hashicorp/setup-terraform](#)

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)