

# Using GitHub CLI extensions

## In this article

- About GitHub CLI extensions
- Finding extensions
- Installing extensions
- Viewing installed extensions
- Updating extensions
- Uninstalling extensions

Learn how to use custom extensions written by other GitHub CLI users.

## About GitHub CLI extensions

**Note:** Extensions outside of GitHub Enterprise Cloud and GitHub CLI are not certified by GitHub Enterprise Cloud and are governed by separate terms of service, privacy policy, and support documentation. To mitigate risk when using third-party extensions, audit the source code of the extension before installing or updating the extension.

GitHub CLI extensions are custom GitHub CLI commands that anyone can create and use. For more information about how to create GitHub CLI extensions, see "[Creating GitHub CLI extensions](#)."

Extensions are locally installed and are scoped to the user. Therefore, if you access GitHub CLI from a different machine or another user accesses GitHub CLI from the same machine, the extension will not be available.

## Finding extensions

You can find extensions by browsing [repositories with the gh-extension topic](#).

## Installing extensions

To install an extension, use the `extensions install` subcommand. Replace the `repo` parameter with the repository of the extension. You can use the full URL, such as `https://github.com/octocat/gh-whoami`, or just the owner and repository, such as `octocat/gh-whoami`.

If the owner and repository are used, `gh` will install the extension using the hostname to which `gh` is currently authenticated. The full URL format is useful when installing extensions from a different host. For example, users on GitHub Enterprise Server should use the full repository URL to install extensions from GitHub.com or any other host.

To install an extension in development from the current directory, use `.` as the value for the `repo` parameter.

```
gh extension install REPO
```

If you already have an extension by the same name installed, the command will fail. For example, if you have installed `octocat/gh-whoami`, you must uninstall it before installing `hubot/gh-whoami`.

## Viewing installed extensions [↗](#)

---

To view all installed extensions, use the `extensions list` subcommand. The output will also tell you which extensions have updates available.

```
gh extension list
```

## Updating extensions [↗](#)

---

To update an extension, use the `extensions upgrade` subcommand. Replace the `extension` parameter with the name of the extension.

```
gh extension upgrade EXTENSION
```

To update all installed extensions, use the `--all` flag.

```
gh extension upgrade --all
```

## Uninstalling extensions [↗](#)

---

To uninstall an extension, use the `extensions remove` subcommand. Replace the `extension` parameter with the name of the extension.

```
gh extension remove EXTENSION
```

### Legal

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