

# Automatically generated release notes

## In this article

About automatically generated release notes

Creating automatically generated release notes for a new release

Configuring automatically generated release notes

Further reading

You can automatically generate release notes for your GitHub releases

### Who can use this feature

Repository collaborators and people with write access to a repository can generate and customize automated release notes for a release.

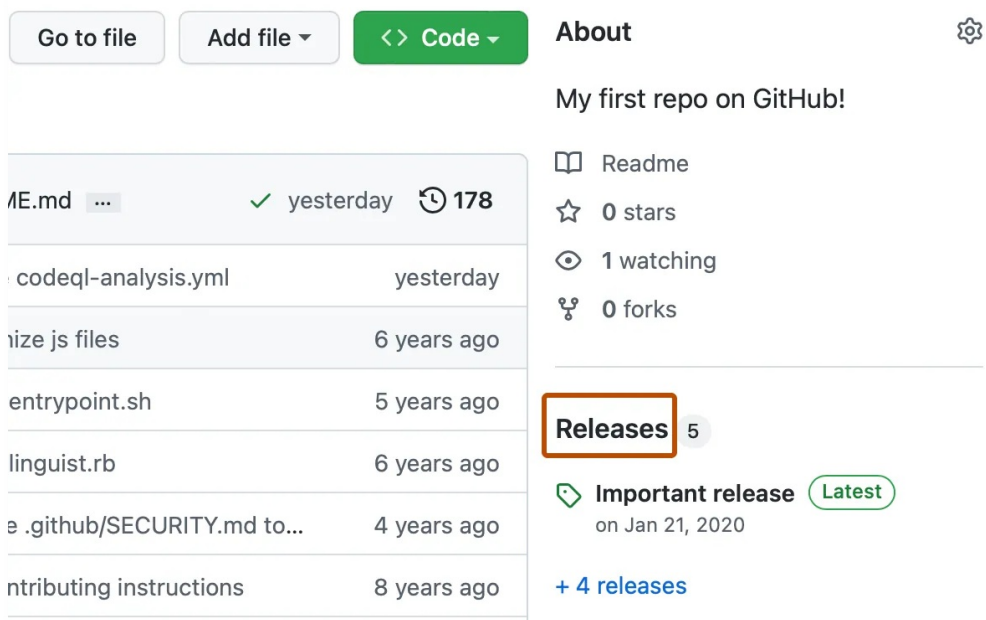
## About automatically generated release notes

Automatically generated release notes provide an automated alternative to manually writing release notes for your GitHub releases. With automatically generated release notes, you can quickly generate an overview of the contents of a release. Automatically generated release notes include a list of merged pull requests, a list of contributors to the release, and a link to a full changelog.

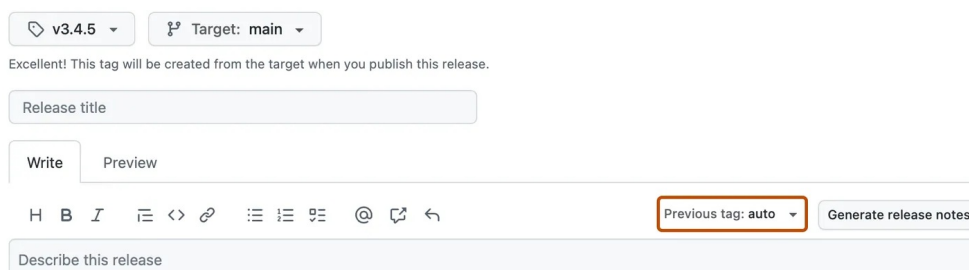
You can also customize your automated release notes, using labels to create custom categories to organize pull requests you want to include, and exclude certain labels and users from appearing in the output.

## Creating automatically generated release notes for a new release

- 1 On your GitHub Enterprise Server instance, navigate to the main page of the repository.
- 2 To the right of the list of files, click **Releases**.



- 3 At the top of the page, click **Draft a new release**.
- 4 To choose a tag for the release, select the **Choose a tag** dropdown menu.
  - To use an existing tag, click the tag.
  - To create a new tag, type a version number for your release, then click **Create new tag**.
- 5 If you created a new tag, select the **Target** dropdown menu, then click the branch that contains the project you want to release.
- 6 Optionally, above the description field, select the **Previous tag** dropdown menu, then click the tag that identifies the previous release.



- 7 In the "Release title" field, type a title for your release.
- 8 Above the description field, click **Generate release notes**.
- 9 Check the generated notes to ensure they include all (and only) the information you want to include.
- 10 Optionally, to include binary files such as compiled programs in your release, drag and drop or manually select files in the binaries box.
- 11 Optionally, to notify users that the release is not ready for production and may be unstable, select **This is a pre-release**.
- 12 Optionally, select **Set as latest release**. If you do not select this option, the latest release label will automatically be assigned based on semantic versioning.
- 13 Optionally, if GitHub Discussions is enabled for the repository, create a discussion

for the release.

- Select **Create a discussion for this release**.
- Select the **Category** dropdown menu, then click a category for the release discussion.

14 If you're ready to publicize your release, click **Publish release**. To work on the release later, click **Save draft**.

## Configuring automatically generated release notes


- 1 On your GitHub Enterprise Server instance, navigate to the main page of the repository.
- 2 Above the list of files, using the **Add file** drop-down, click **Create new file**.
- 3 In the file name field, type `.github/release.yml`. This will create a new file called `release.yml` in the `.github` directory.
- 4 In the file, using the configuration options below, specify in YAML the pull request labels and authors you want to exclude from this release. You can also create new categories and list the pull request labels to be included in each of them.

### Configuration options

Parameter	Description
<code>changelog.exclude.labels</code>	A list of labels that exclude a pull request from appearing in release notes.
<code>changelog.exclude.authors</code>	A list of user or bot login handles whose pull requests are to be excluded from release notes.
<code>changelog.categories[*].title</code>	<b>Required.</b> The title of a category of changes in release notes.
<code>changelog.categories[*].labels</code>	<b>Required.</b> Labels that qualify a pull request for this category. Use <code>*</code> as a catch-all for pull requests that didn't match any of the previous categories.
<code>changelog.categories[*].exclude.labels</code>	A list of labels that exclude a pull request from appearing in this category.
<code>changelog.categories[*].exclude.authors</code>	A list of user or bot login handles whose pull requests are to be excluded from this category.

### Example configurations

A configuration for a repository that labels semver releases

YAML 

```
# .github/release.yml

changelog:
```

```
exclude:
  labels:
    - ignore-for-release
authors:
  - octocat
categories:
  - title: Breaking Changes
    labels:
      - Semver-Major
      - breaking-change
  - title: Exciting New Features
    labels:
      - Semver-Minor
      - enhancement
  - title: Other Changes
    labels:
      - "*"

```

A configuration for a repository that doesn't tag pull requests but where we want to separate out Dependabot automated pull requests in release notes ( `labels: '*'` is required to display a catchall category)

YAML



```
# .github/release.yml

changelog:
  categories:
    - title: Features
      labels:
        - '*'
    exclude:
      labels:
        - dependencies
  - title: Dependencies
    labels:
      - dependencies

```

## Further reading

- ["Managing labels"](#)

### Legal

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