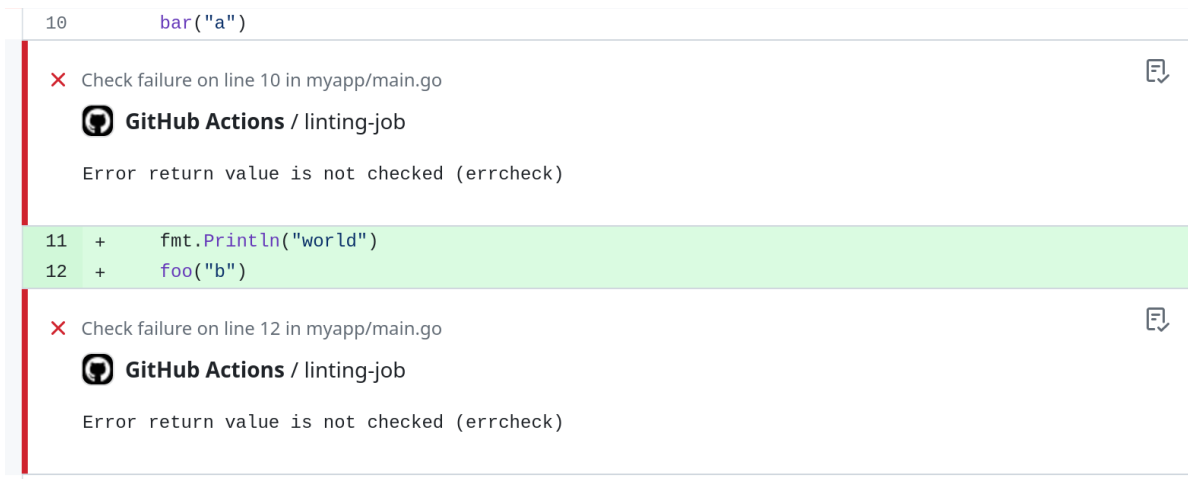

golangci-lint-action



It's the official GitHub action for golangci-lint from its authors.

The action runs golangci-lint and reports issues from linters.



Compatibility

- **v5.0.0+** removes `skip-pkg-cache` and `skip-build-cache` because the cache related to Go itself is already handled by `actions/setup-go`.
- **v4.0.0+** requires an explicit `actions/setup-go` installation step before using this action: `uses: actions/setup-go@v5`. The `skip-go-installation` option has been removed.
- **v2.0.0+** works with `golangci-lint` version `>= v1.28.3`
- **v1.2.2** is deprecated due to we forgot to change the minimum version of `golangci-lint` to `v1.28.3` (issue)
- **v1.2.1** works with `golangci-lint` version `>= v1.14.0` (issue)

How to use

Add `.github/workflows/golangci-lint.yml` with the following contents:

```
1 name: golangci-lint
2 on:
3   push:
4     branches:
```

```

5     - main
6     - master
7     pull_request:
8
9     permissions:
10    contents: read
11    # Optional: allow read access to pull request. Use with `only-new-
12    #   issues` option.
13    # pull-requests: read
14
15    jobs:
16      golangci:
17        name: lint
18        runs-on: ubuntu-latest
19        steps:
20          - uses: actions/checkout@v4
21          - uses: actions/setup-go@v5
22            with:
23              go-version: '1.21'
24          - name: golangci-lint
25            uses: golangci/golangci-lint-action@v5
26            with:
27              # Require: The version of golangci-lint to use.
28              # When `install-mode` is `binary` (default) the value can be
29              #   v1.2 or v1.2.3 or `latest` to use the latest version.
30              # When `install-mode` is `goinstall` the value can be v1.2.3,
31              #   `latest`, or the hash of a commit.
32              version: v1.57
33
34              # Optional: working directory, useful for monorepos
35              # working-directory: somedir
36
37              # Optional: golangci-lint command line arguments.
38              #
39              # Note: By default, the `.golangci.yml` file should be at the
40              #   root of the repository.
41              # The location of the configuration file can be changed by
42              #   using `--config=`
43              # args: --timeout=30m --config=/my/path/.golangci.yml --
44              #   issues-exit-code=0
45
46              # Optional: Show only new issues.
47              # If you are using `merge_group` event (merge queue) you
48              #   should add the option `fetch-depth: 0` to `actions/
49              #   checkout` step.
50              # The default value is `false`.
51              # only-new-issues: true
52
53              # Optional: if set to true, then all caching functionality
54              #   will be completely disabled,
55              #   takes precedence over all other caching options.

```

```
47         # skip-cache: true
48
49         # Optional: if set to true, caches will not be saved, but
50         #           they may still be restored,
51         #           subject to other options
52         # skip-save-cache: true
53
54         # Optional: The mode to install golangci-lint. It can be '
55         #           binary' or 'goinstall'.
56         # install-mode: "goinstall"
```

We recommend running this action in a job separate from other jobs (go test, etc.) because different jobs run in parallel.

Multiple OS Support

If you need to run linters for specific operating systems, you will need to use the action [>=v2](#).

Here is a sample configuration file:

```
1  name: golangci-lint
2  on:
3    push:
4      branches:
5        - main
6        - master
7    pull_request:
8
9  permissions:
10   contents: read
11   # Optional: allow read access to pull request. Use with `only-new-
12   #           issues` option.
13   # pull-requests: read
14
15  jobs:
16    golangci:
17      strategy:
18        matrix:
19          go: ['1.21']
20          os: [ubuntu-latest, macos-latest, windows-latest]
21      name: lint
22      runs-on: ${ matrix.os }
23      steps:
24        - uses: actions/checkout@v4
25        - uses: actions/setup-go@v5
26          with:
27            go-version: ${ matrix.go }
28        - name: golangci-lint
29          uses: golangci/golangci-lint-action@v5
```

```

29     with:
30         # Require: The version of golangci-lint to use.
31         # When `install-mode` is `binary` (default) the value can be
32         #   v1.2 or v1.2.3 or `latest` to use the latest version.
33         # When `install-mode` is `goinstall` the value can be v1.2.3,
34         #   `latest`, or the hash of a commit.
35         version: v1.57
36
37         # Optional: working directory, useful for monorepos
38         # working-directory: somedir
39
40         # Optional: golangci-lint command line arguments.
41         #
42         # Note: By default, the `.golangci.yml` file should be at the
43         #   root of the repository.
44         # The location of the configuration file can be changed by
45         #   using `--config=`
46         # args: --timeout=30m --config=/my/path/.golangci.yml --
47         #   issues-exit-code=0
48
49         # Optional: Show only new issues.
50         # If you are using `merge_group` event (merge queue) you
51         #   should add the option `fetch-depth: 0` to `actions/
52         #   checkout` step.
53         # The default value is `false`.
54         # only-new-issues: true
55
56         # Optional: if set to true, then all caching functionality
57         #   will be completely disabled,
58         #   takes precedence over all other caching options.
59         # skip-cache: true
60
61         # Optional: if set to true, caches will not be saved, but
62         #   they may still be restored,
63         #   subject to other options
64         # skip-save-cache: true
65
66         # Optional: The mode to install golangci-lint. It can be '
67         #   binary' or 'goinstall'.
68         # install-mode: "goinstall"

```

You will also likely need to add the following `.gitattributes` file to ensure that line endings for Windows builds are properly formatted:

```
1 *.go text eol=lf
```

Comments and Annotations

Currently, GitHub parses the action's output and creates annotations.

The restrictions of annotations are the following:

1. Currently, they don't support Markdown formatting (see the feature request)
2. They aren't shown in the list of comments. If you would like to have comments - please, up-vote the issue.

To enable annotations, you need to add the 'checks' permission to your action.

```
yaml annotate permissions: # Required: allow read access to the
content for analysis. contents: read # Optional: allow read access
to pull request. Use with `only-new-issues` option. pull-requests:
read # Optional: Allow write access to checks to allow the action to
annotate code in the PR. checks: write
```

Performance

The action was implemented with performance in mind:

1. We cache data from golangci-lint analysis between builds by using @actions/cache.
2. We don't use Docker because image pulling is slow.
3. We do as much as we can in parallel, e.g. we download cache, and golangci-lint binary in parallel.

For example, in a repository of golangci-lint running this action without the cache takes 50s, but with cache takes 14s: * in parallel: * 4s to restore 50 MB of cache * 1s to find and install `golangci-lint` * 1s to run `golangci-lint` (it takes 35s without cache)

Internals

We use JavaScript-based action. We don't use Docker-based action because:

1. Docker pulling is slow currently
2. it's easier to use caching from @actions/cache

We support different platforms, such as `ubuntu`, `macos`, and `windows` with `x32` and `x64` archs.

Inside our action, we perform 3 steps:

1. Setup environment running in parallel:
 - restore cache of previous analyses
 - fetch action config and find the latest `golangci-lint` patch version for needed version (users of this action can specify only minor version of `golangci-lint`). After that install `golangci-lint` using @actions/tool-cache

-
2. Run `golangci-lint` with specified by user `args`
 3. Save cache for later builds

Caching internals

1. We save and restore the following directory: `~/.cache/golangci-lint`.
2. The primary caching key looks like `golangci-lint.cache-{interval_number}-{go.mod_hash}`. Interval number ensures that we periodically invalidate our cache (every 7 days). `go.mod` hash ensures that we invalidate the cache early - as soon as dependencies have changed.
3. We use restore keys: `golangci-lint.cache-{interval_number}-`. GitHub matches keys by prefix if we have no exact match for the primary cache.

This scheme is basic and needs improvements. Pull requests and ideas are welcome.

Development of this action

1. Install act
2. Make a symlink for `act` to work properly: `ln -s . golangci-lint-action`
3. Prepare deps once: `npm run prepare-deps`
4. Run `npm run local` after any change to test it