golangci-lint Go linters runner

CONCEPTS

One binary runs many linters in parallel over a shared parse, with a cache. Exit code 1 means issues were found.

Packages ./...
↓ load
Linters default + enable
↓ report
Filters exclusions, nolint, new-from
↓ print
Output text, json, sarif...

Coming from v1? Config needs version: "2", exclude rules moved to linters.exclusions, formatters (gofmt, goimports) have their own section. Run golangci-lint migrate.

INSTALL AND INFO

curl -sSfL https://golangci-lint.run/install.sh | sh -s -- -b $(go env GOPATH)/bin v*<version>*
Install a pinned release binary

golangci-lint --version
Show version and Go it was built with

golangci-lint help linters
Describe every available linter

RUN

golangci-lint run
Lint all packages below here

golangci-lint run *<dir>*/...
Lint one subtree

golangci-lint run --fix
Apply fixes linters can make

golangci-lint run --timeout 5m
Abort after a time limit

golangci-lint run --tests=false
Skip *_test.go files

golangci-lint run --build-tags *<tag>*
Lint with build tags set

golangci-lint run -c *<file>*
Use a specific config file

golangci-lint run -v
Show config used, linters and timings

RUN (cont.)

golangci-lint cache clean
Clear the analysis cache

PICK LINTERS

golangci-lint linters
List enabled and disabled linters

golangci-lint run -E *<linter>*,*<linter>*
Enable extra linters

golangci-lint run -D *<linter>*
Disable a linter

golangci-lint run --enable-only *<linter>*
Run just this linter

golangci-lint run --default all
Start from every linter

golangci-lint run --default none -E *<linter>*
Start empty, add only these

golangci-lint run --fast-only
Keep only fast linters

The standard set (default)

Linter	Catches
errcheck	unchecked errors
govet	go vet checks
staticcheck	bugs, simplifications, style
unused	unused code
ineffassign	assignments never read

ONLY NEW ISSUES

golangci-lint run --new-from-rev *<rev>*
Issues introduced since a commit

golangci-lint run --new-from-merge-base main
Issues since the branch left main

golangci-lint run -n
Issues in uncommitted changes

OUTPUT

golangci-lint run --output.json.path stdout
Machine-readable JSON

golangci-lint run --max-issues-per-linter 0 --max-same-issues 0
Show every issue, no caps

FORMAT

golangci-lint fmt
Format files with enabled formatters

golangci-lint fmt -d
Show diff, change nothing

golangci-lint fmt -E goimports
Format with a given formatter

CONFIG FILE

Searched as .golangci.yml/.yaml/.toml/.json from the linted path up to /, then in $HOME.

.golangci.yml

```yaml
version: "2"
linters:
  default: standard
  enable: [gosec, revive, misspell]
  disable: [errcheck]
  settings:
    misspell:
      locale: US
  exclusions:
    generated: lax
    presets: [std-error-handling]
    paths: [third_party/]
    rules:
      - path: _test\.go
        linters: [gosec]
formatters:
  enable: [gofumpt, goimports]
run:
  timeout: 5m
```

golangci-lint config verify
Validate config against the schema

golangci-lint config path
Which config file is used

golangci-lint migrate
Convert a v1 config to v2

SUPPRESS

//nolint:*<linter>*
Silence a linter on this line

//nolint:*<linter>* // *<why>*
Silence with a reason

//nolint:*<a>*,*<b>*
Silence several linters

A //nolint line above a func, var block or package clause covers that whole declaration or file.