

REFERÊNCIA RÁPIDA GITHUB CLI

Repositórios, issues, PRs, actions, releases, API

Configuração

Instalação

```
brew install gh          macOS via Homebrew
sudo apt install gh     Debian / Ubuntu
winget install GitHub.cli Windows via winget
conda install gh        Via conda-forge
```

Autenticação

```
gh auth login           # interactive login
gh auth login --with-token < token.txt
gh auth status         # check auth state
gh auth refresh -s repo,gist # add scopes
```

Configuração

```
gh config set editor vim
gh config set pager less
gh config set git_protocol ssh
gh config list
```

Repositórios

Comandos de Repositório

```
gh repo create my-app --public --clone
gh repo clone owner/repo
gh repo fork owner/repo --clone
gh repo view owner/repo --web
```

Opções de Repositório

```
--public | --private      Definir visibilidade do repositório
--template owner/repo     Criar a partir de repositório template
```

```
--clone                  Clonar após criar
--add-readme             Inicializar com README
gh repo list owner       Listar repositórios do owner
gh repo delete owner/repo Excluir repositório (com confirmação)
gh repo rename new-name Renomear repositório atual
gh repo archive owner/repo Arquivar um repositório
```

Issues

Gerenciando Issues

```
gh issue create --title "Bug" --body "Details here"
gh issue list --state open --label bug
gh issue view 42
gh issue close 42 --reason completed
```

Opções de Issue

```
--assignee @me          Atribuir a si mesmo
--label bug,urgent      Adicionar labels
--milestone v2.0        Definir milestone
--project "Board"       Adicionar ao projeto
gh issue edit 42        Editar issue interativamente
gh issue reopen 42      Reabrir issue fechada
gh issue comment 42 -b "msg" Adicionar comentário à issue
gh issue pin 42         Fixar issue no repositório
```

Pull Requests

Criar e Gerenciar PRs

```
gh pr create --title "feat: add auth" --body "... "
gh pr create --fill # title/body from commits
gh pr list --state open
gh pr view 123 --web
```

Revisar e Fazer Merge

```
gh pr checkout 123 # check out PR branch
gh pr diff 123 # view PR diff
gh pr review 123 --approve
gh pr merge 123 --squash --delete-branch
```

Opções de PR

```
--draft                Criar como PR draft
--reviewer user1,user2 Solicitar revisores
--base main            Definir branch base
--merge | --squash | --rebase Estratégia de merge
--auto                Habilitar auto-merge quando checks passam
--delete-branch        Excluir branch após merge
gh pr ready 123        Marcar PR draft como pronto
```

Actions

Comandos de Workflow

```
gh run list           # recent runs
gh run view 12345 # run details
gh run view 12345 --log-failed # failed step logs
gh run watch 12345 # live status
```

Acionar e Gerenciar

```
gh workflow run deploy.yml --ref main
gh workflow list
gh workflow view deploy.yml
gh run rerun 12345 --failed # rerun failed jobs
```

Opções de Actions

```
-f key=value          Passar input para workflow_dispatch
--json              Saída como JSON
-b branch            Filtrar por branch
gh run download 12345 Baixar artefatos da execução
gh cache list        Listar caches do Actions
gh cache delete KEY Excluir uma entrada de cache
```

Releases

Gerenciando Releases

```
gh release create v1.0.0 --generate-notes
gh release create v1.0.0 ./dist/*.tar.gz
gh release list
gh release view v1.0.0
```

Opções de Release

```
--title "Release v1.0" Definir título da release
--notes "Changelog here" Definir notas de release inline
```

```
--notes-file CHANGELOG.md Notas de arquivo
--generate-notes          Geral automaticamente a partir de commits
--draft                  Criar como draft
--pre-release            Marcar como pré-release
--latest                 Marcar como release mais recente
gh release download v1.0.0 Baixar assets da release
gh release delete v1.0.0 Excluir uma release
gh release edit v1.0.0 Editar metadados da release
```

Gists

Comandos de Gist

```
gh gist create file.py -d "My snippet"
gh gist create file1.js file2.js # multi-file gist
gh gist list
gh gist view <id>
```

Opções de Gist

```
-d "description" Definir descrição do gist
--public         Criar gist público (padrão: secreto)
--web           Abrir gist no navegador
gh gist edit <id> Editar arquivos do gist
gh gist clone <id> Clonar gist localmente
gh gist delete <id> Excluir um gist
```

API

Fazendo Chamadas de API

```
gh api repos/owner/repo
gh api repos/owner/repo/issues --method POST \
-f title="Bug" -f body="Details"
gh api graphql -f query='{ viewer { login } }'
```

Formatando Saída

```
gh api repos/owner/repo --jq '.stargazers count'
gh api repos/owner/repo --template '{{.full_name}}'
gh pr list --json number,title --jq '.[].title'
```

Opções de API

```
--method GET|POST|PUT|DELETE Método HTTP
-f key=value Definir campo string
-F key=@file Definir campo a partir de arquivo
--jq 'expression' Filtrar JSON com sintaxe jq
--template 'tmpl' Formatar com template Go
--paginate Buscar todas as páginas
-H 'Accept: ...' Definir header personalizado
```

Aliases

Gerenciando Aliases

```
gh alias set co 'pr checkout'
gh alias set bugs 'issue list --label bug'
gh alias set last 'run list -L 1'
gh alias list
```

Aliases Avançados

```
# Alias with shell command
gh alias set --shell pv 'gh pr view --json url --jq .url | pbcopy'
```

Excluir Aliases

```
# Delete alias
gh alias delete co
```

Dicas de Alias

```
gh alias set name 'cmd' Criar alias simples
--shell Alias executa via shell (suporta pipes)
gh alias list Mostrar todos os aliases definidos
gh alias delete name Remover um alias
```

Padrões Comuns

Fluxo Diário

```
gh issue list --assignee @me # my open issues
gh pr status # PRs needing attention
gh pr checks 123 # CI status for PR
gh run list -b main -L 5 # recent CI runs
```

Buscando

```
gh search repos --language rust --stars ">1000"
gh search issues --repo owner/repo "memory leak"
gh search prs --state open --review required
```

Flags Úteis

```
--json field1,field2 Saída de campos específicos como JSON
--jq 'expr' Filtrar saída JSON
-L N Limitar resultados a N itens
--web Abrir resultado no navegador
-R owner/repo Direcionar para repositório específico
GH_TOKEN=xxx Auth via variável de ambiente
```