

Riferimento Rapido GitHub CLI

Repository, issue, PR, actions, release, API

Configurazione

Installazione

```
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
```

Autenticazione

```
gh auth login           # login interattivo
gh auth login --with-token < token.txt
gh auth status          # controlla stato auth
gh auth refresh -s repo,gist # aggiungi scope
```

Configurazione

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

Repository

Comandi Repository

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

Opzioni Repository

```
--public | --private      Imposta la visibilità del repository
--template owner/repo     Crea da repository template
--clone                   Clona dopo la creazione
--add-readme              Inizializza con README
gh repo list owner        Elenca repo per owner
gh repo delete owner/repo Elimina repository (con conferma)
gh repo rename new-name  Rinomina il repo corrente
gh repo archive owner/repo Archivia un repository
```

Issue

Gestire le Issue

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

Opzioni Issue

```
--assignee @me           Assegna a te stesso
--label bug,urgent       Aggiunge etichette
--milestone v2.0         Imposta milestone
--project "Board"        Aggiunge al progetto
gh issue edit 42         Modifica issue interattivamente
gh issue reopen 42       Riapre una issue chiusa
gh issue comment 42 -b "msg" Aggiunge commento alla issue
gh issue pin 42          Fissa la issue al repo
```

Pull Request

Creazione e Gestione PR

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

Revisione e Merge

```
gh pr checkout 123      # checkout branch PR
gh pr diff 123          # visualizza diff PR
gh pr review 123 --approve
gh pr merge 123 --squash --delete-branch
```

Opzioni PR

```
--draft                  Crea come draft PR
--reviewer user1,user2  Richiede revisori
--base main              Imposta branch base
--merge | --squash | --rebase Strategia di merge
--auto                  Abilita auto-merge quando i check passano
--delete-branch         Elimina branch dopo il merge
gh pr ready 123         Marca draft PR come pronta
```

Actions

Comandi Workflow

```
gh run list             # esecuzioni recenti
gh run view 12345      # dettagli esecuzione
gh run view 12345 --log-failed # log step falliti
gh run watch 12345    # stato in tempo reale
```

Avvio e Gestione

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

Opzioni Actions

```
-f key=value            Passa input a workflow_dispatch
--json                 Output come JSON
-b branch              Filtra per branch
gh run download 12345  Scarica artifact dell'esecuzione
gh cache list          Elenca le cache di Actions
gh cache delete KEY   Elimina una voce cache
```

Release

Gestire le Release

```
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
```

Opzioni Release

```
--title "Release v1.0" Imposta il titolo della release
--notes "Changelog"    Imposta le note inline
--notes-file CHANGELOG.md Note da file
--generate-notes        Genera automaticamente dai commit
--draft                 Crea come bozza
--prerelease            Marca come prerelease
--latest                Marca come ultima release
gh release download v1.0.0 Scarica asset della release
gh release delete v1.0.0 Elimina una release
gh release edit v1.0.0  Modifica metadati release
```

Gist

Comandi Gist

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

Opzioni Gist

```
-d "description"        Imposta la descrizione del gist
--public                Crea gist pubblico (default: segreto)
--web                   Apri gist nel browser
gh gist edit <id>      Modifica file del gist
gh gist clone <id>     Clona il gist localmente
gh gist delete <id>   Elimina un gist
```

API

Chiamate API

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

Formattazione Output

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

Opzioni API

```
--method GET|POST|PUT|DELETE Metodo HTTP
-f key=value                    Imposta campo stringa
-F key=@file                    Imposta campo da file
--jq 'expression'              Filtra JSON con sintassi jq
--template 'tmpl'              Formatta con template Go
--paginate                     Scarica tutte le pagine
-H 'Accept: ...'               Imposta header personalizzato
```

Alias

Gestire gli Alias

```
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
```

Alias Avanzati

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

# Elimina alias
gh alias delete co
```

Consigli sugli Alias

```
gh alias set name 'cmd'      Crea alias semplice
--shell                      L'alias viene eseguito via shell (supporta pipe)
gh alias list                 Mostra tutti gli alias definiti
gh alias delete name         Rimuove un alias
```

Pattern Comuni

Workflow Quotidiano

```
gh issue list --assignee @me # mie issue aperte
gh pr status                 # PR che richiedono attenzione
gh pr checks 123            # stato CI per PR
gh run list -b main -L 5    # esecuzioni CI recenti
```

Ricerca

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

Riferimento Rapido GitHub CLI

Flag Utili

<code>--json field1,field2</code>	Output campi specifici come JSON
<code>--jq 'expr'</code>	Filtra output JSON
<code>-L N</code>	Limita risultati a N elementi
<code>--web</code>	Apri risultato nel browser
<code>-R owner/repo</code>	Punta a un repository specifico
<code>GH_TOKEN=xxx</code>	Auth via variabile d'ambiente