

RÉFÉRENCE RAPIDE GITHUB CLI

Dépôts, issues, PRs, actions, releases, API

Configuration

Installation

```
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
```

Authentification

```
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
```

Configuration

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

Dépôts

Commandes de dépôt

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

Options de dépôt

```
--public | --private    Définir la visibilité du dépôt
--template owner/repo   Créer depuis un dépôt modèle
--clone                 Cloner après la création
--add-readme            Initialiser avec un README
gh repo list owner      Lister les dépôts du propriétaire
gh repo delete owner/repo Supprimer le dépôt (avec confirmation)
gh repo rename new-name Renommer le dépôt courant
gh repo archive owner/repo Archiver un dépôt
```

Issues

Gérer les 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
```

Options d'issue

```
--assignee @me          S'assigner à soi-même
--label bug,urgent      Ajouter des labels
--milestone v2.0        Définir un jalon
--project "Board"       Ajouter à un projet
gh issue edit 42        Modifier une issue de façon interactive
gh issue reopen 42      Rouvrir une issue fermée
gh issue comment 42 -b "msg" Ajouter un commentaire à une issue
gh issue pin 42         Épingler une issue dans le dépôt
```

Pull Requests

Créer et gérer des 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
```

Réviser et fusionner

```
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
```

Options de PR

```
--draft                Créer en tant que PR brouillon
--reviewer user1,user2 Demander des relecteurs
--base main            Définir la branche de base
--merge | --squash | --rebase Stratégie de fusion
--auto                Activer la fusion automatique quand les vérifications passent
--delete-branch        Supprimer la branche après la fusion
gh pr ready 123        Marquer une PR brouillon comme prête
```

Actions

Commandes 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
```

Déclencher et gérer

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

Options d'actions

```
-f key=value           Passer une entrée à workflow_dispatch
--json                Sortie au format JSON
--b branch            Filtrer par branche
gh run download 12345 Télécharger les artefacts d'un run
gh cache list          Lister les caches Actions
gh cache delete KEY   Supprimer une entrée de cache
```

Releases

Gérer les 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
```

Options de release

```
--title "Release v1.0" Définir le titre de la release
--notes "Changelog here" Définir les notes de release en ligne
--notes-file CHANGELOG.md Notes depuis un fichier
--generate-notes        Générer automatiquement depuis les commits
--draft                Créer en tant que brouillon
--pre-release          Marquer comme pré-release
--latest               Marquer comme dernière release
gh release download v1.0.0 Télécharger les assets de la release
gh release delete v1.0.0 Supprimer une release
gh release edit v1.0.0 Modifier les métadonnées de la release
```

Gists

Commandes 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>
```

Options de gist

```
-d "description"       Définir la description du gist
--public              Créer un gist public (défaut : secret)
--web                 Ouvrir le gist dans le navigateur
gh gist edit <id>     Modifier les fichiers du gist
gh gist clone <id>   Cloner le gist localement
gh gist delete <id>  Supprimer un gist
```

API

Appels 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 } }'
```

Formatage de la sortie

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

Options API

```
--method GET|POST|PUT|DELETE Méthode HTTP
-f key=value                 Définir un champ de type chaîne
-F key=@file                 Définir un champ depuis un fichier
--jq 'expression'           Filtrer le JSON avec la syntaxe jq
--template 'tmpl'          Formater avec un template Go
--paginate                  Récupérer toutes les pages
-H 'Accept: ...'           Définir un en-tête personnalisé
```

Alias

Gérer les 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 avancés

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

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

Conseils sur les alias

```
gh alias set name 'cmd' Créer un alias simple
--shell                L'alias s'exécute via shell (supporte les pipes)
gh alias list          Afficher tous les alias définis
gh alias delete name   Supprimer un alias
```

Patterns courants

Workflow quotidien

```
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
```

Recherche

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

Indicateurs utiles

```
--json field1,field2 Sortir des champs spécifiques en JSON
--jq 'expr'           Filtrer la sortie JSON
-L N                  Limiter les résultats à N éléments
--web                 Ouvrir le résultat dans le navigateur
--R owner/repo       Cibler un dépôt spécifique
GH_TOKEN=xxx         Authentification via variable d'environnement
```