

git-pr

Description

`git-pr` is the top-level command for interacting with [pull requests](#) on a source code hosting platform like [GitLab](#) or [GitHub](#). The top-level command `git-pr` doesn't do anything by itself, instead it delegates all functionality to multiple sub-commands. The available sub-commands to `git-pr` are:

- [apply](#)
- [approve](#)
- [checkout](#)
- [close](#)
- [create](#)
- [fetch](#)
- [info](#)
- [integrate](#)
- [list](#)
- [reviewer](#)
- [set](#)
- [show](#)
- [sponsor](#)
- [test](#)

For more information about a sub-command, see its own wiki page.

All sub-commands takes an optional pull request id as their first argument. The pull request is optional because if you create a pull request with `git pr create` then `git-pr` remembers the local branch you created the pull request from. This means you don't have to specify the id for subsequent operations such as `git pr set` or `git pr integrate`.

Usage

```
$ git pr -h
git-pr is used for interacting with pull requests from a command line.
The following sub-commands are available:

- apply -- apply a pull request
  Usage: git pr apply [<ID>]
  Flags:
    -u, --username NAME  Username on host
    -r, --remote NAME    Name of remote, defaults to 'origin'
    --no-token           Do not use a personal access token (PAT)
    --verbose           Turn on verbose output
    --debug             Turn on debugging output
    --version           Print the version of this tool

- approve -- approve a pull request
  Usage: git pr approve [<ID>]
  Flags:
    -u, --username NAME  Username on host
    -r, --remote NAME    Name of remote, defaults to 'origin'
    -m, --message TEXT   Message to author as part of approval (e.g. "Looks good!")
    --verbose           Turn on verbose output
    --debug             Turn on debugging output
    --version           Print the version of this tool

- cc -- add one or more labels
  Usage: git pr cc <ARG>...
  Flags:
    --verbose           Turn on verbose output
    --debug             Turn on debugging output
    --version           Print the version of this tool

- checkout -- checkout a pull request
  Usage: git pr checkout [<ID>]
  Flags:
    -u, --username NAME  Username on host
    -r, --remote NAME    Name of remote, defaults to 'origin'
    -b, --branch NAME    Create a new branched named NAME
    --no-token           Do not use a personal access token (PAT)
    --verbose           Turn on verbose output
    --debug             Turn on debugging output
```

```

--version      Print the version of this tool

- close -- close a pull request
Usage: git pr close [<ID>]
Flags:
  -u, --username NAME  Username on host
  -r, --remote NAME     Name of remote, defaults to 'origin'
  --verbose            Turn on verbose output
  --debug             Turn on debugging output
  --version           Print the version of this tool

- contributor -- add or remove contributors
Usage: git pr contributor [<ID>]
Flags:
  --add USERNAME      Consider pull request contributed to by this user
  --remove USERNAME   Do not consider pull request contributed to by this user
  --verbose           Turn on verbose output
  --debug            Turn on debugging output
  --version          Print the version of this tool

- create -- create a pull request
Usage: git pr create [<ID>]
Flags:
  -u, --username NAME      Username on host
  -r, --remote NAME        Name of remote, defaults to 'origin'
  -b, --branch NAME        Name of target branch, defaults to 'master'
  --cc MAILING LISTS       Mailing lists to CC for initial RFR e-mail
  --backport REV           Create a backport pull request for the given revision
  --ignore-workspace       Ignore local changes in worktree and staging area when creating pull request
  --ignore-local-commits   Ignore local commits not pushed when creating pull request
  --publish                Publish the local branch before creating the pull request
  --jcheck                 Run jcheck before creating the pull request
  --draft                  Create a pull request in draft state
  --verbose                Turn on verbose output
  --debug                  Turn on debugging output
  --version                Print the version of this tool

- csr -- require CSR for the pull request
Usage: git pr csr [<ID>]
Flags:
  --needed      This pull request needs an approved CSR before integration
  --unneeded    This pull request does not need an approved CSR before integration
  --verbose     Turn on verbose output
  --debug      Turn on debugging output
  --version    Print the version of this tool

- fetch -- fetch a pull request
Usage: git pr fetch [<ID>]
Flags:
  -u, --username NAME  Username on host
  -r, --remote NAME     Name of remote, defaults to 'origin'
  --no-token           Do not use a personal access token (PAT)
  --verbose            Turn on verbose output
  --debug             Turn on debugging output
  --version           Print the version of this tool

- info -- show status of a pull request
Usage: git pr info [<ID>]
Flags:
  -u, --username NAME  Username on host
  -r, --remote NAME     Name of remote, defaults to 'origin'
  --no-decoration       Hide any decorations when listing PRs
  --no-token            Do not use a personal access token (PAT)
  --checks              Show information about checks
  --author              Show the author of the pull request
  --title               Show the title of the pull request
  --assignees           Show the assignees of the pull request
  --reviewers           Show the reviewers of the pull request
  --contributors        Show the additional contributors to the pull request
  --issues              Show the issues associated with the pull request
  --commits             Show the commits in pull request

```

```

--branch          Show the target branch for the pull request
--url             Show the url for the pull request
--status          Show the status for the pull request
--labels          Show the labels for the pull request
--verbose         Turn on verbose output
--debug           Turn on debugging output
--version         Print the version of this tool

- integrate -- integrate a pull request
Usage: git pr integrate [<ID>]
Flags:
  -u, --username NAME Username on host
  -r, --remote NAME   Name of remote, defaults to 'origin'
  --atomic             Integrate the pull request atomically
  --verbose           Turn on verbose output
  --debug             Turn on debugging output
  --version           Print the version of this tool

- issue -- add, remove or create issues
Usage: git pr issue [<ID>]
Flags:
  --add ID            Consider issue solved by this pull request
  --remove ID         Do not consider issue as solved by this pull request
  --priority 1|2|3|4|5 Priority for issue
  --component NAME   Component for issue
  --create            Create an issue for this pull request
  --verbose           Turn on verbose output
  --debug            Turn on debugging output
  --version           Print the version of this tool

- list -- list open pull requests
Usage: git pr list [<ID>]
Flags:
  -u, --username NAME Username on host
  -r, --remote NAME   Name of remote, defaults to 'origin'
  --authors LIST      Comma separated list of authors
  --assignees LIST    Comma separated list of assignees
  --labels LIST       Comma separated list of labels
  --issues LIST       Comma separated list of issues
  --columns id,title,author,assignees,labels Comma separated list of columns to show
  --no-decoration     Hide any decorations when listing PRs
  --no-draft          Hide all pull requests in draft state
  --no-token          Do not use a personal access token (PAT)
  --verbose           Turn on verbose output
  --debug            Turn on debugging output
  --version           Print the version of this tool

- reviewer -- add or remove reviewers
Usage: git pr reviewer [<ID>]
Flags:
  --add USERNAME     Consider pull request reviewed by this user
  --remove USERNAME  Do not consider pull request reviewed by this user
  --verbose           Turn on verbose output
  --debug            Turn on debugging output
  --version           Print the version of this tool

- set -- set properties of a pull request
Usage: git pr set [<ID>]
Flags:
  -u, --username NAME Username on host
  -r, --remote NAME   Name of remote, defaults to 'origin'
  --assignees LIST    Comma separated list of assignees
  --title MESSAGE     The title of the pull request
  --open             Set the pull request's state to open
  --closed           Set the pull request's state to closed
  --body             Set the body of the pull request
  --no-draft         Mark the pull request as not draft
  --verbose           Turn on verbose output
  --debug            Turn on debugging output
  --version           Print the version of this tool

```

```
- show -- show a pull request
Usage: git pr show [<ID>]
Flags:
  -u, --username NAME  Username on host
  -r, --remote NAME    Name of remote, defaults to 'origin'
  --no-token           Do not use a personal access token (PAT)
  --verbose            Turn on verbose output
  --debug              Turn on debugging output
  --version            Print the version of this tool

- sponsor -- sponsor a pull request
Usage: git pr sponsor [<ID>]
Flags:
  -u, --username NAME  Username on host
  -r, --remote NAME    Name of remote, defaults to 'origin'
  --verbose            Turn on verbose output
  --debug              Turn on debugging output
  --version            Print the version of this tool

- summary -- add a summary to the commit message for the pull request
Usage: git pr summary [<ID>]
Flags:
  --remove            Remove an existing summary
  --verbose           Turn on verbose output
  --debug             Turn on debugging output
  --version           Print the version of this tool

- test -- test a pull request
Usage: git pr test [<ID>]
Flags:
  -u, --username NAME  Username on host
  -r, --remote NAME    Name of remote, defaults to 'origin'
  -j, --job NAME       Name of a job
  --approve            Approve a test request
  --verbose           Turn on verbose output
  --debug             Turn on debugging output
  --version           Print the version of this tool
```

Configuration

There is no configuration for the top-level git-pr command, but many of the sub-commands can be configured. See each sub-command's wiki page for more information on how configure a particular sub-command.

Source

Please see [GitPr.java](#).