

Gip

gip contributors

Version 0.5.1

Usage

Gip offers bulk management for your Git projects running Git commands on a set of repositories.

Repositories can be registered in a YAML or JSON file.

Fields are:

- **name**: a descriptive name for the project
- **repository**: the remote repository url
- **local_path**: local directory where operate into
- **pull_policy**: behaviour when pull command is ran

YAML example:

```
- name: gip
  repository: "https://github.com/enr/gip.git"
  local_path: /tmp/gip
- name: nowhere
  repository: "https://nowhere.git"
  local_path: /nowhere/not/found
  pull_policy: never
```

JSON example:

```
[
  {
    "name": "gip",
    "repository": "https://github.com/enr/gip.git",
    "local_path": "/tmp/gip"
  },
  {
    "name": "nowhere",
    "repository": "https://nowhere.git",
    "local_path": "/nowhere/not/found",
    "pull_policy": "never"
  }
]
```

You can set the path using the **-f** flag:

```
gip -f examples/linux.json -d pull -a
```

Otherwise it defaults to the file **.gip** in your home directory.

Available commands

list

List all managed projects

```
gip -f examples/linux.json -d ls
```

pull

Run git pull on managed projects.

The default behaviour is to pull only projects which **local_path** exists.

Projects with **pull_policy** set to **never** will be skipped, projects with **pull_policy** set to **always** will be pulled even if local path does not exist.

Use **never** for repositories that you don't want to pull (for example because they are reachable only using VPN).

```
gip -f examples/linux.json -d pull
```

status

Run **git status** on all managed projects.

statusfull shows untracked files too.

```
gip -f examples/linux.json -d status
```

Get status hiding the warn about missing local directory:

```
gip -m sf
```

other options

Verbose output:

```
$ gip -d sf
```

Choose a path other than **~/.gip** for configuration file:

```
$ gip -f ~/.config/local/gip.json -d sf
```

Install

The latest release is in: <https://github.com/enr/gip/releases/latest>

Put **gip** in your **\$PATH** and make it executable:

```
$ curl -sL https://github.com/enr/gip/releases/download/v0.4.8/gip-0.4.8_linux_amd64.zip -o gip-0.4.8_linux_amd64.zip

$ unzip gip-0.4.8_linux_amd64.zip
Archive:  gip-0.4.8_linux_amd64.zip
  creating: gip-0.4.8_linux_amd64/
  inflating: gip-0.4.8_linux_amd64/gip

$ ./gip-0.4.8_linux_amd64/gip --version
gip version 0.4.8
Revision: 359819e2a81d7b5bc827af1adec84eb42632f706
Build date: 2020-02-05T22:30:46Z

$ ./gip-0.4.8_linux_amd64/gip --help
NAME:
  gip - Keep tracks of your Git projects

USAGE:
  gip [global options] command [command options] [arguments...]

VERSION:
  0.4.8
Revision: 359819e2a81d7b5bc827af1adec84eb42632f706
Build date: 2020-02-05T22:30:46Z

COMMANDS:
  status, s
  statusfull, sf
  list, ls
  pull
  help, h      Shows a list of commands or help for one command

GLOBAL OPTIONS:
  --file value, -f value  path to the configuration file to use (if not set will be
picked ${HOME}/.gip)
  --debug, -d             operates in debug mode: lot of output (default: false)
  --quiet, -q             operates in quiet mode (default: false)
  --ignore-missing, -m    ignores missing local directories, otherwise prints a warn
(default: false)
  --help, -h              show help (default: false)
  --version, -v           print the version (default: false)
```

License

Copyright © 2020 gip contributors.