

Collaborative Software Development with Git and GitHub

CoDaS-HEP school (2019) at Princeton University

David Luet

Princeton University
PICSciE, Research Computing/OIT, Department of Geosciences

July 25, 2019

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

References

Conclusion

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

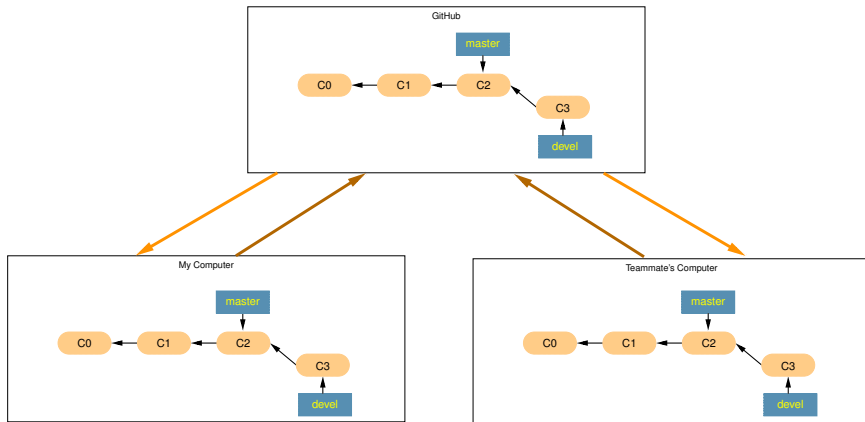
Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

A Simple Collaborative Workflow

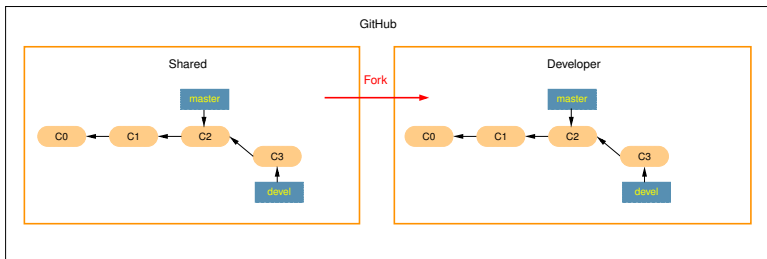


Remote Repository Permissions

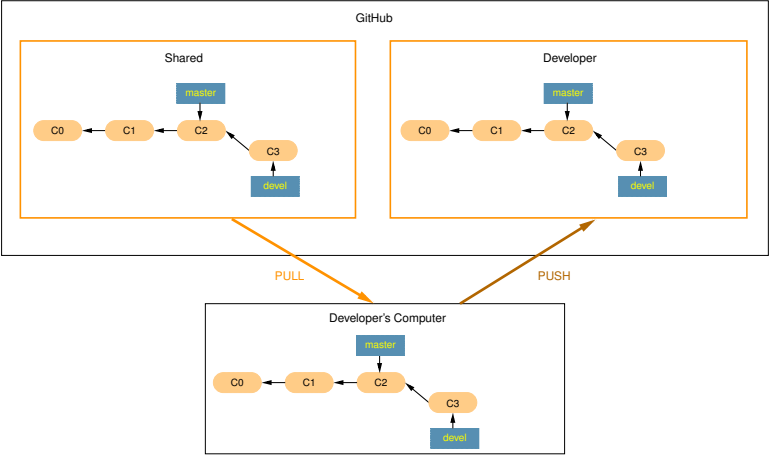
- ▶ Problem with that simple workflow:
 - ▶ It's easy to push **broken code** to the shared repository.
 - ▶ when others pull the changes and start adding their development, it can create problems.
 - ▶ this does not work well with more than two developers.
- ▶ To solve this issue, we introduce two roles with different permissions to the shared repository:
 - ▶ code **maintainers**: **push and pull** permissions.
 - ▶ **developers**: **only pull** permission.

Forking a Repository on GitHub

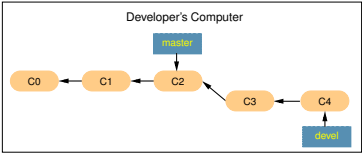
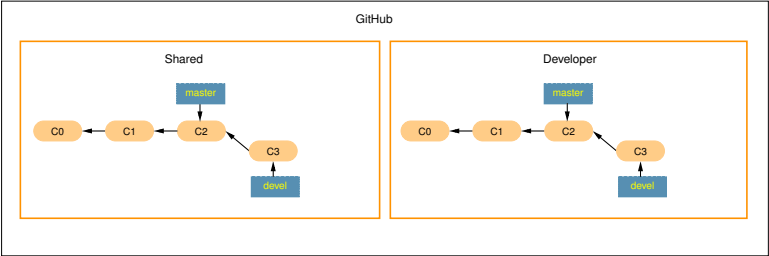
Forks are basically a copy of a repo on GitHub.



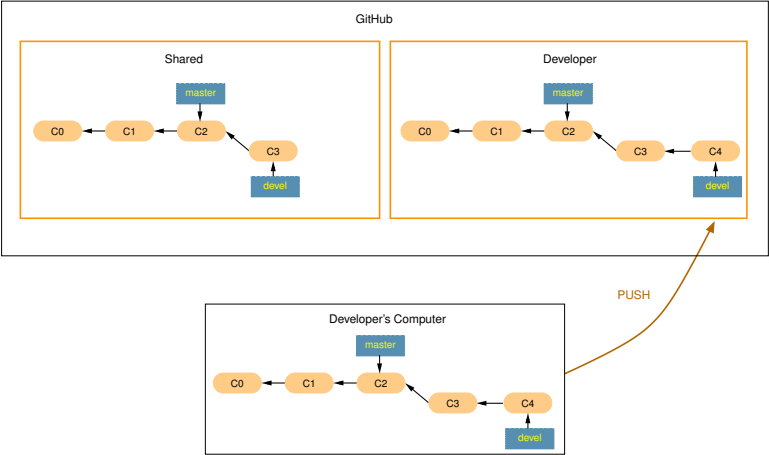
Pull-Request: the Different Repositories



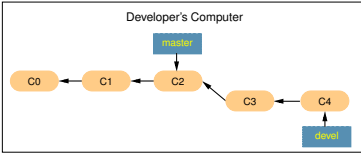
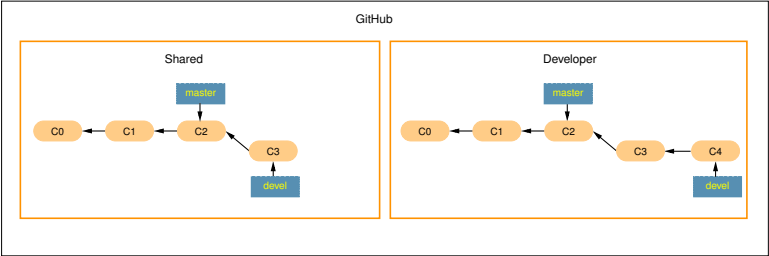
Pull-Request Steps



Pull-Request Steps



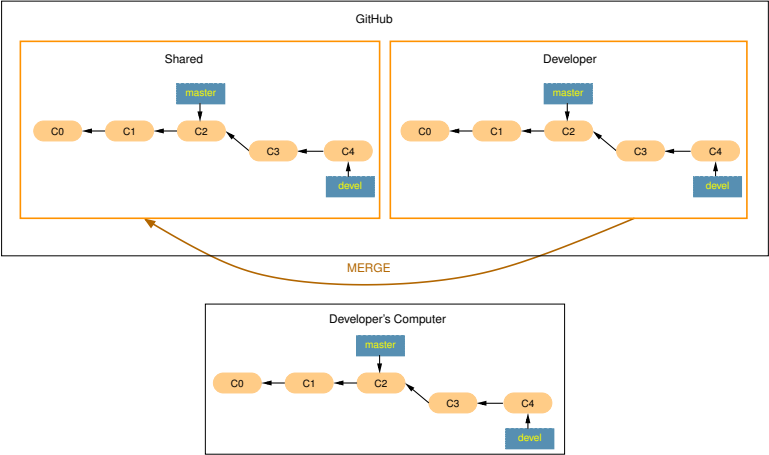
Pull-Request Steps



Open Pull-Request

GitHub

Pull-Request Steps



Advantages of Doing a Pull-Request?

- ▶ Gives us time to **review** and test the changes before committing them.
- ▶ So that **no broken code** gets committed to the shared repository.
- ▶ The problem with this simple workflow is that it can be hard for the code maintainers to know whether or not changes break the code.
- ▶ That's why we need to build some **tests**.

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Download files and create repo locally

- ▶ Download from
<https://github.com/luet/factorial/releases>
- ▶ untar
- ▶ build repo locally:

```
$ git init  
$ git add -A  
$ git commit -m "Initial commit"
```

Push on GitHub

- ▶ Go on GitHub, create a new repo, make sure it's public.
- ▶ Add the remote:

- ▶ If you have ssh keys set up:

```
$ git remote add origin \  
    git@github.com:[your_GitHub_id]/factorial.git
```

- ▶ Otherwise use the https URL.

- ▶ Push to the repo:

```
git push -u origin master
```

GitHub repo 1

The screenshot shows a GitHub repository page for 'buildbot-princeton...' with a sidebar on the left containing 'Repositories' and 'Your teams' sections. A central modal window is open, titled 'Learn Git and GitHub without any code!', with a green 'Read the guide' button and a white 'Start a project' button. Below the modal, a notification states 'You've been added to the PrincetonUniversity organization!' with a 'defunkt' profile picture. At the bottom, a commit message is visible: '1 commit to feat/composition/backprop_3DC_converging_pathways'.

buildbot-princeton... ▾

Repositories New

Find a repository...

[murthylab/sleep](#)

[geodynamics/spectem3d_globe](#)

[geodynamics/spectem2d](#)

[geodynamics/spectem3d](#)

[PrincetonUniver... /PsyNeuLink](#)

[PrincetonUniver... /athena](#)

[brainiak/brainiak](#)

Show more

Your teams

Find a team...

[intelPNI/builders](#)

[PrincetonUniversity/david_team_4](#)

[PrincetonUniversity/djfr_temp4](#)

[PrincetonUniversity/a_new_team_for_he...](#)

[geodynamics/spectem-developers](#)

×

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)

×

You've been added to the **PrincetonUniversity** organization!

Here are some quick tips for a first-time organization member.

- Use the switch context button in the upper left corner of this page to switch between your personal context (**buildbot-princeton**) and organizations you are a member of.
- After you switch contexts you'll see an organization-focused dashboard that lists out organization repositories and activities.

defunkt ▾

jdcpci pushed to PrincetonUniversity/PsyNeuLink 2 hours ago

1 commit to [feat/composition/backprop_3DC_converging_pathways](#)

[scattop](#) • [tests/composition/test_learning](#)

GitHub repo 2

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner

 buildbot-princeton ▾

Repository name *

factorial ✓

Great repository names are short and memorable. Need inspiration? How about [verbose-octo-enigma](#)?

Description (optional)

 **Public**

Anyone can see this repository. You choose who can commit.

 **Private**

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾



Add a license: **None** ▾



Create repository

GitHub repo 3

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# factorial" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:buildbot-princeton/factorial.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:buildbot-princeton/factorial.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

 **ProTip!** Use the URL for this page when adding GitHub as a remote.

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Developer forks repo 1

- ▶ From GitHub go to the web site for the Maintainers web site:
<https://github.com/buildbot-princeton/factorial>

Developer forks repo 2

buildbot-princeton / factorial

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights

No description, website, or topics provided.

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

luet Initial commit

Latest commit 582f181 19 minutes ago

.gitignore

Initial commit

19 minutes ago

README.md

Initial commit

19 minutes ago

factorial.py

Initial commit

19 minutes ago

test.py

Initial commit

19 minutes ago

README.md

This is small Python script that computes the [factorial](#) of an integer.

- Clone this code with:

```
git clone https://github.com/PrincetonUniversity/jenkins_tutorial.git
```

- This code is written for Python 3, on Princeton University clusters the default Python is a version 2. So, to get this to work on say adroit, you need to load an environment module:

```
$ module load anaconda3
```

Developer forks repo 3

Y djifr / **factorial**

forked from buildbot-princeton/factorial

👁 Watch 0

★ Star 0

🍴 Fork 1

↔ Code

↑ Pull requests 0

📁 Projects 0

📖 Wiki

🔒 Security

📊 Insights

⚙ Settings

Forking buildbot-princeton/factorial

It should only take a few seconds.

🔄 Refresh



Developer clones the fork locally

```
git clone git@github.com:djlfjr/factorial.git
```

The screenshot shows the GitHub interface for a repository named 'factorial' by user 'djlfjr'. The repository is a fork of 'buildbot-princeton/factorial'. At the top, there are statistics for Watch (0), Star (0), and Fork (1). Below this is a navigation bar with links for Code, Pull requests (0), Projects (0), Wiki, Security, Insights, and Settings. A message states 'No description, website, or topics provided.' with an 'Edit' button. Below this, it shows '1 commit', '1 branch', '0 releases', and '1 contributor'. A 'Clone or download' button is highlighted in green. A dropdown menu is open, showing options to 'Clone with SSH' (selected), 'Use HTTPS', 'Open in Desktop', and 'Download ZIP'. The repository files list includes: .gitignore (Initial commit), README.md (Initial commit), factorial.py (Initial commit), and test.py (Initial commit). The README.md file is open, showing a description of a Python script that computes the factorial of an integer. It includes a list item 'Clone this code with:' followed by a code block containing the command: `git clone https://github.com/PrincetonUniversity/jenkins_tutorial.git`. Another list item states: 'This code is written for Python 3, on Princeton University clusters the default Python is a version 2. So, to get this to'.

Developer sets up the local clone so that it can be sync'ed with upstream

- ▶ **Set a remote:**

```
git remote add upstream git@github.com:buildbot-princ
```

- ▶ `git fetch --all`

- ▶ **check diff:**

```
git diff origin/master  
git diff upstream/master
```


Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Developer makes a change to the code 1

- ▶ **Create a branch:**

```
$ git checkout -b trouble_maker  
Switched to a new branch 'trouble_maker'
```

- ▶ **Make a change that will break the test.**
- ▶ **commit the change and push it to the Fork:**

```
git add factorial.py  
git commit -m "Trouble"  
git push origin trouble_maker
```

Developer makes a change to the code 2

Look at the web site:

The screenshot shows a GitHub repository page for 'factorial' by user 'djifr'. The repository is forked from 'buildbot-princeton/factorial'. At the top, there are navigation links for 'Code', 'Pull requests', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The repository has 2 commits, 2 branches, 0 releases, and 2 contributors. A recently pushed branch named 'trouble_maker' is highlighted in yellow, with a 'Compare & pull request' button. Below this, there are buttons for 'Branch: trouble_maker', 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. The main content area shows a comparison between the 'trouble_maker' branch and the 'master' branch. It lists files: '.gitignore', 'README.md', 'factorial.py', and 'test.py', along with their commit history. The 'README.md' file is expanded to show its content: 'This is small Python script that computes the factorial of an integer.' and a list item 'Clone this code with:'.

Factorial by **djifr** forked from buildbot-princeton/factorial

Watch 0 Star 0 Fork 1

Code Pull requests 0 Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

2 commits 2 branches 0 releases 2 contributors

Your recently pushed branches:

trouble_maker (1 minute ago) [Compare & pull request](#)

Branch: **trouble_maker** [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

This branch is 1 commit ahead of buildbot-princeton:master. [Pull request](#) [Compare](#)

djifr <small>Trouble</small>	Latest commit 3159442 3 minutes ago
.gitignore	Initial commit 2 hours ago
README.md	Initial commit 2 hours ago
factorial.py	Trouble 3 minutes ago
test.py	Initial commit 2 hours ago

README.md [Edit](#)

This is small Python script that computes the **factorial** of an integer.

- Clone this code with:

Developer makes a change to the code 3

- ▶ **switch to branch** `trouble_maker`

This branch is 1 commit ahead of `buildbot-princeton:m`

- ▶ **Also:**

```
$ git remote update
```

```
$ git log --graph --abbrev-commit --oneline
```

```
* 31594d2 (HEAD -> trouble_maker, origin/trouble_maker)
```

```
* 502f181 (upstream/master, origin/master, origin/HEAD)
```

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Developer opens a Pull Request 1

- ▶ because Developer doesn't have write access.
- ▶ Click Pull request

The screenshot shows the GitHub interface for the repository `buildbot-princeton/factorial`. At the top, there are navigation links for `Code`, `Issues`, `Pull requests`, `Projects`, `Wiki`, `Security`, and `Insights`. The `Pull requests` link is highlighted. Below the navigation, the heading `Open a pull request` is displayed, followed by a subtext: `Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.`

The main form for creating a pull request is visible, with the following fields:

- `base repository:` `buildbot-princeton/factorial`
- `base:` `master`
- `head repository:` `djifr/factorial`
- `compare:` `trouble_maker`

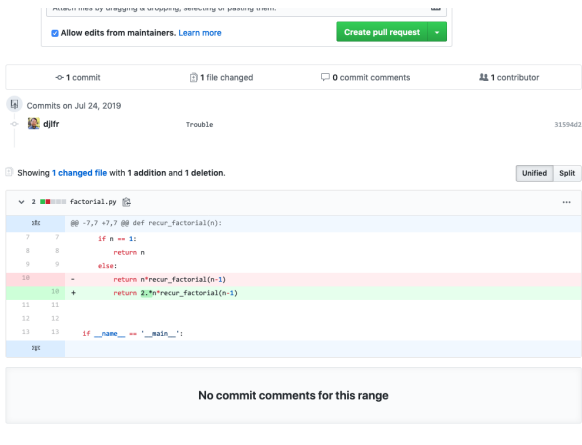
Below the form, a green checkmark indicates: `✓ Able to merge. These branches can be automatically merged.`

The pull request title is `Trouble`. The description area is empty, with a `Write` tab selected and a `Preview` tab. A rich text editor toolbar is visible above the description area. Below the description area, there is a text input for `Leave a comment` and a note: `Attach files by dragging & dropping, selecting or pasting them.`

At the bottom of the form, there is a checkbox for `Allow edits from maintainers. Learn more` and a green `Create pull request` button.

At the bottom of the page, a summary bar shows: `1 commit`, `1 file changed`, `0 commit comments`, and `1 contributor`.

Developer opens a Pull Request 2



Allow edits from maintainers. [Learn more](#) Create pull request

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Jul 24, 2019

djlftr Trouble 31594d2

Showing 1 changed file with 1 addition and 1 deletion. Unified Split

```
2 factorial.py @@ -7,7 +7,7 @@ def recur_factorial(n):
7     if n == 1:
8         return n
9     else:
10    -     return n*recur_factorial(n-1)
11    +     return 2*n*recur_factorial(n-1)
12
13 if __name__ == '__main__':
```

No commit comments for this range

- ▶ Click Create pull request

Developer opens a Pull Request 3

It takes you to the Maintainers repo:

The screenshot shows a GitHub Pull Request page for the repository 'buildbot-princeton / factorial'. The pull request is titled 'Trouble #1' and is in the 'Open' state. It shows that 'djifr' wants to merge 1 commit into the 'buildbot-princeton:master' branch from the 'djifr:trouble_maker' branch. The interface includes navigation tabs for Code, Issues, Pull requests, Projects, Wiki, Security, and Insights. A comment by 'djifr' is visible, stating 'No description provided.' Below the comment, a green box indicates that the branch has no conflicts with the base branch. At the bottom, there is a 'Write' section with a text area for a comment and a rich text editor toolbar. On the right side, there are sections for 'Reviewers', 'Assignees', 'Labels', 'Projects', 'Milestone', and 'Notifications'. The 'Notifications' section shows a button to 'Unsubscribe' and a note that the user is receiving notifications because they authored the thread.

buildbot-princeton / factorial

Watch 0 Star 0 Fork 1

Code Issues Pull requests Projects Wiki Security Insights

Trouble #1

Edit

Open djifr wants to merge 1 commit into buildbot-princeton:master from djifr:trouble_maker

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

djifr commented now

No description provided.

Trouble 31594d2

Add more commits by pushing to the `trouble_maker` branch on `djifr/factorial`.

This branch has no conflicts with the base branch
Only those with [write access](#) to this repository can merge pull requests.

Write Preview

AA B i « > ↻ ☰ ☷ ⌨ 📎 ↶ ↷

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Reviewers
No reviews

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications [Customize](#)

[Unsubscribe](#)

You're receiving notifications because you authored the thread.

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Maintainer looks at the Pull Request

buildbot-princeton / factorial

Watch 0 Star 0 Fork 1

Code Issues 0 Pull requests 1 Projects 0 Wiki Security Insights Settings

Label issues and pull requests for new contributors

[Dismiss](#)

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [help wanted](#) or [good first issue](#)

Filters

Labels 0 Milestones 0

New pull request

1 Open 0 Closed

Author ▾

Labels ▾

Projects ▾

Milestones ▾

Reviews ▾

Assignee ▾

Sort ▾

Trouble

#1 opened 1 minute ago by djfr

ProTip! Updated in the last three days: [updated->2019-07-21](#).



Maintainer test Pull Request

- ▶ Maintainer clones the Developer's branch locally and runs the tests:

```
$ git clone https://github.com/dj1lfr/factorial.git \  
    --single-branch --branch trouble_maker
```

```
$ cd factorial
```

```
$ python test.py
```

```
F
```

```
=====
```

```
FAIL: test_value (__main__.TestReturnValues)
```

```
-----
```

```
Traceback (most recent call last):
```

```
  File "test.py", line 9, in test_value
```

```
    self.assertEqual(recur_factorial(3), 6)
```

```
AssertionError: 24.0 != 6
```

```
-----
```

```
Ran 1 test in 0.000s
```

```
FAILED (failures=1)
```

Maintainer comments on the PR that the test has failed

The screenshot shows a GitHub Pull Request (PR) interface. At the top, it says "Open" and "djlfr wants to merge 1 commit into buildbot-princeton:master from djlfr:trouble_maker". Below this, there are statistics: Conversation 0, Commits 1, Checks 0, and Files changed 1. A green checkmark icon indicates that the branch has no conflicts with the base branch.

Comments from maintainers:

- A comment from "djlfr" (10 minutes ago) says "No description provided."
- A comment from "buildbot-princet..." (commented now) says "Your changes fail our test, please correct your mistake."

A green box with a checkmark icon contains the following text:

This branch has no conflicts with the base branch
Merging can be performed automatically.

Below this, there is a green button labeled "Merge pull request" and a link to "You can also open this in GitHub Desktop or view command line instructions."

At the bottom, there is a "Write" section with a "Preview" tab and a text area for leaving a comment. The text area contains the placeholder text "Leave a comment" and a note "Attach files by dragging & dropping, selecting them."

On the right side, there are several settings sections:

- Reviewers:** No reviews.
- Assignees:** No one—assign yourself.
- Labels:** None yet.
- Projects:** None yet.
- Milestone:** No milestone.
- Notifications:** Customize. A button labeled "Unsubscribe" is visible.
- Participants:** 2 participants.
- Lock conversation:** A lock icon and the text "Lock conversation".

- ▶ **notice:** Add more commits by pushing to the trouble_maker branch on djlfr/factorial.
- ▶ as long as the PR is opened, you can add commits.

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Developer fixes the mistake and pushes again

```
$ git branch
  master
  * trouble_maker
$ # edit factorial.py
$ git add factorial.py
$ git commit -m "Fix error"
$ git push
```

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Maintainer pulls the new commit and test again

```
$ git remote -v
origin https://github.com/djlfactorial.git (fetch)
origin https://github.com/djlfactorial.git (push)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reuse=0
Unpacking objects: 100% (3/3), done.
From https://github.com/djlfactorial
   31594d2..44d923e  trouble_maker -> origin/trouble_maker
Updating 31594d2..44d923e
Fast-forward
 factorial.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```


Maintainer pulls the new commit and test again

```
$ python test.py
```

```
.
```

```
-----  
Ran 1 test in 0.000s
```

```
OK
```

Maintainer merges the Pull Request

buildbot-princet... commented 11 minutes ago Owner + @ ...

Your changes fail our test, please correct your mistake.

Fix error 44d923e

Add more commits by pushing to the `trouble_maker` branch on `djffr/factorial`.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request ▼ You can also open [this in GitHub Desktop](#) or view [command line instructions](#).

✓ Create a merge commit
All commits from this branch will be added to the base branch via a merge commit.

Squash and merge
The 2 commits from this branch will be combined into one commit in the base branch.

Rebase and merge
The 2 commits from this branch will be rebased and added to the base branch.

Close pull request Comment

Labels ⚙
None yet

Projects ⚙
None yet

Milestone ⚙
No milestone

Notifications Customize
Unsubscribe
You're receiving notifications because you commented.

2 participants

Lock conversation

ProTip! Add comments to specific lines under [Files changed](#).



Maintainer merges the Pull Request

- ▶ Multiple merge strategies (see [About merge methods on GitHub - GitHub Help](#))
- ▶ Here we choose to create a merge commit:

The screenshot shows the GitHub interface for the repository `buildbot-princeton / factorial`. At the top, there are buttons for `Watch` (0), `Star` (0), and `Fork` (1). Below this is a navigation bar with links for `Code`, `Issues` (0), `Pull requests` (0), `Projects` (0), `Wiki`, `Security`, `Insights`, and `Settings`. A dropdown menu shows the current branch is `master`.

The main content area is titled "Commits on Jul 24, 2019" and displays a list of commits. The top commit is a "Merge pull request #1 from djifr/trouble_maker", which is marked as "Verified" and has a commit hash of `6b4cbb9`. Below this are three individual commits from user `djifr`:

- "Fix error" committed 15 minutes ago with hash `44d923e`.
- "Trouble" committed 34 minutes ago with hash `31594d2`.
- "Initial commit" committed 2 hours ago with hash `5b2f181`.

At the bottom of the commit list are buttons for "Newer" and "Older".

The footer contains the copyright notice "© 2019 GitHub, Inc." followed by links for `Terms`, `Privacy`, `Security`, `Status`, and `Help`. On the right side of the footer are links for `Contact GitHub`, `Pricing`, `API`, `Training`, `Blog`, and `About`.

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Developer updates the fork

Update local copy of the remotes:

```
$ git remote update
Fetching origin
Fetching upstream
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reus
Unpacking objects: 100% (1/1), done.
From github.com:buildbot-princeton/factorial
    502f181..6b4cbb9  master      -> upstream/master
```

Developer updates the fork

Sync the master branch with upstream/master

```
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
$ git merge upstream/master
Updating 502f181..6b4cbb9
Fast-forward
 factorial.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Developer deletes the feature branch

Delete the feature branch:

```
$ git branch --merged
* master
  trouble_maker
$ git branch -d trouble_maker
Deleted branch trouble_maker (was 44d923e).
$ git push --delete origin trouble_maker
To github.com:djlfjr/factorial.git
- [deleted]          trouble_maker
```

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

References

- ▶ GitHub:
 - ▶ [Fork a repo - GitHub Help](#)
 - ▶ [Configuring pull request merges - GitHub Help](#)
 - ▶ [About merge methods on GitHub - GitHub Help](#)
- ▶ Atlassian (Bitbucket):
 - ▶ [Collaborating in general](#)
 - ▶ [Comparing Workflows](#)
 - ▶ [Original blog post by Vincent Driessen](#)

Outline

Motivation: Different roles and pull request

Maintainer Builds Repo

Developer Forks Repo

Developer Makes a Change

Developer Opens a Pull Request

Maintainer Inspects Pull Request

Developer Fixes Mistake

Maintainer Inspects Pull Request

Developer Updates the Fork and Deletes Feature Branch

Conclusion

- ▶ Git and GitHub are powerful tools for collaborative software development.
- ▶ Main features:
 - ▶ Provides a reliable way to exchange code developments.
 - ▶ Keeps track of who has done what.
 - ▶ Through multiple permissions settings and the pull request mechanism, it gives some level of protection to the code.
- ▶ An essential part of collaborative development is testing.
- ▶ Automatic testing: Travis, Jenkins, buildbot.