

🔗 **ai2-education-fiep-turma-4 / git-katas** Publicforked from [eficode-academy/git-katas](#)**Code**

Pull requests

Actions

Wiki

Security

Insights

🔗 master ▾

...

git-katas / basic-stashing /

This branch is 1 commit ahead of eficode-academy:master.

🔗 Contribute ▾

**JKrag** ...

on 15 Apr 2020



..



README.md

2 years ago



setup.ps1

3 years ago



setup.sh

2 years ago

Git Kata: Basic stashing

Setup:

☰ README.md

The task

You are working on your project. You've staged some work and have some unstaged work as well. Suddenly, you're made aware that a bug has made it to production. You'll stash your work, fix the bug and get back to your original work.

1. Explore the repo
 - i. What work do you have in the working directory?
 - ii. What work do you have staged ?

iii. What does the commit log look like ?

Notice that file.txt has some staged changes (i.e. changes in the index) and unstaged changes (changes in the working directory)

2. Use `git stash` to stash your current work.

i. Now, what work do you have in the working directory?

ii. What work do you have staged ?

iii. What does the commit log look like ?

iv. What does the stash list look like ?

3. Fix the typos in bug.txt on master and commit your changes.

4. Now to get back to your work, apply the stash to master.

i. What work do you have in the working directory?

ii. What work do you have staged ?

Oops. All our changes are unstaged now. This may be undesirable and unexpected

5. Undo our changes with `git reset --hard HEAD`. This is an unsafe command as it will remove files from your index and working directory permanently, but we have our changes safely stashed so we're ok. Review the [reset](#) kata if you're unsure of what happens here.

6. Apply the stash to master with the `--index` option.

i. What work do you have in the working directory?

ii. What work do you have staged ?

Ok, back to where we were!

7. We won't need the stash anymore. Drop it.

i. What does the stash list look like ?

ii. What does the commit log look like ?

Useful commands

- `git status`
- `git status -s`
- `git diff`
- `git diff master`
- `git stash`
- `git stash list`
- `git stash apply`
- `git stash apply --index`
- `git stash drop`
- `git log --oneline --all --graph`

- `git commit`
 - `git add`
-