

GITHUB Exercise

Main goal

The course is aimed to obtain the basic principles of the Git and GitHub VCS. At the end all the students would get the public account on the GitHub service (if not persisted before), fork the external project, create a branch, commit a refinement with tag and create a pull request.

Initial state

Every student is supplied with the credentials for the virtual course PC and could reach it via HTTP + SSH.

Preliminary items

1. Install the Git on the course environment.

To install the Git on the environment please run the following commands on the command line:

```
apt-get update  
apt-get install git-core
```

Or
yum install -y git-core

Check the Git version:
git --version

2. Configure Git

We need to configure local git by adding the personal credentials, for that please run via the command line interface (CLI) the commands:

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

3. Create the GitHub account (if not persisted before)

To create an account please proceed with the GitHub official portal:

- <https://github.com/join>
- Fill in the Username, Email and Password fields and press the "Create an account" button

Create your personal account

Username

This will be your username. You can add the name of your organization later.

Email address

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

By clicking "Create an account" below, you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

Create an account

Working with Git

1. Create a test repository on the GitHub
Proceed with the link: <https://github.com>, sign in and press "Create a new repository" button:

Repositories	New repository
You don't have any repositories yet!	

Owner: / Repository name: **time-tracker** ✓

Great repository names are short and memorable. Need inspiration? How about **verbose-giggle**.

Description (optional): **Test repository for the JB course**

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

- Clone the persisted test public repository to your environment
To clone the repository please run the commands:

```
cd ~
mkdir time_tracker
cd time_tracker/
git clone https://github.com/zivkashtan/course.git
```

You should see the following:

```
Cloning into 'course'...
remote: Counting objects: 51, done.
remote: Total 51 (delta 0), reused 0 (delta 0), pack-reused 51
Unpacking objects: 100% (51/51), done.
Checking connectivity... done.
```

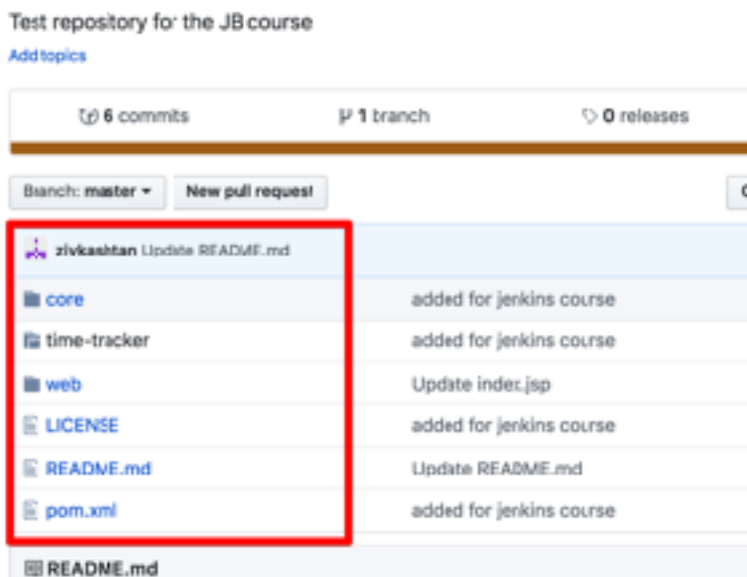
- Push the just cloned test repository to your GitHub
To push the cloned repository we need to remove the old ORIGIN and determine the new ORIGIN (destination). Via the SSH please do:

```
cd ./course
git remote remove origin
git remote add origin https://github.com/<your\_GitHub\_ID>/time-tracker.git
git push --all origin
```

Provide your GitHub login and password. You have to see something like that:

```
Username for 'https://github.com':   
Password for 'https://  
Counting objects: 51, done.  
Delta compression using up to 2 threads.  
Compressing objects: 100% (32/32), done.  
Writing objects: 100% (51/51), 11.02 KiB | 0 bytes/s, done.  
Total 51 (delta 7), reused 0 (delta 0)  
remote: Resolving deltas: 100% (7/7), done.  
To https://github.com/kerldeo/time-tracker.git  
* [new branch]      master -> master
```

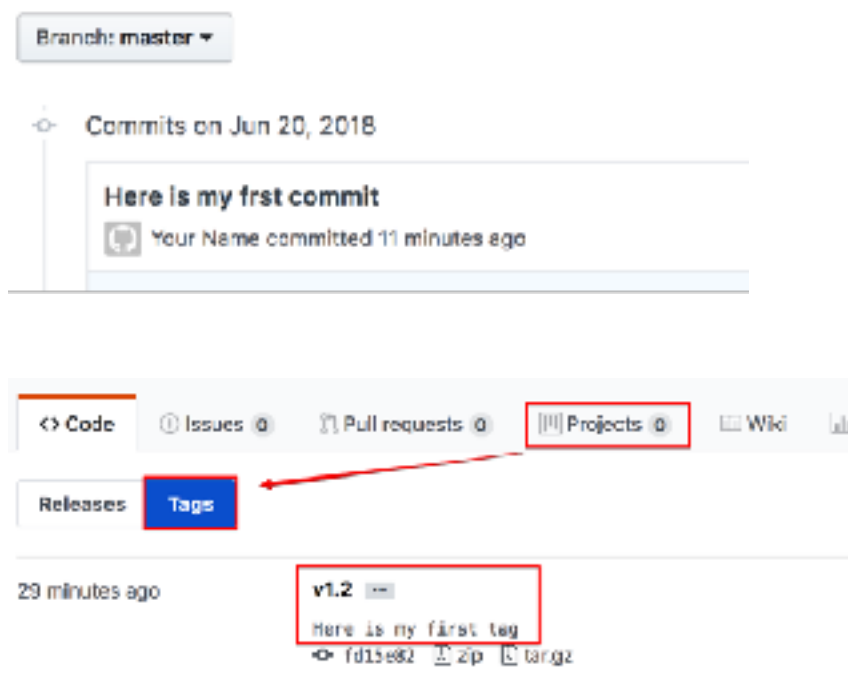
Go to https://github.com/<your_GitHub_ID>/time-tracker and see the cloned repository within the test application:



4. Add the new file under the source control
Here we would create a new test file, put it under the source control and push it to the GitHub with tag. In the CLI please do:

```
cd ~/time_tracker/course  
touch test.txt; echo "I am a Git Overlord there!" > test.txt  
git add test.txt  
git tag -a v1.2 -m 'Here is my first tag'  
git commit -a -m 'Here is my first commit'  
git push --all origin  
git push --tags origin
```

Check the GitHub for the changes:



5. Create a new branch

By that time we have had only one branch named “master”. It is high time to have an extra one. We could create it via the web portal or using the CLI. Let’s start with CLI:

cd ~time_tracker/course

Create a new branch develop:

git checkout -b develop

Check that it has been created:

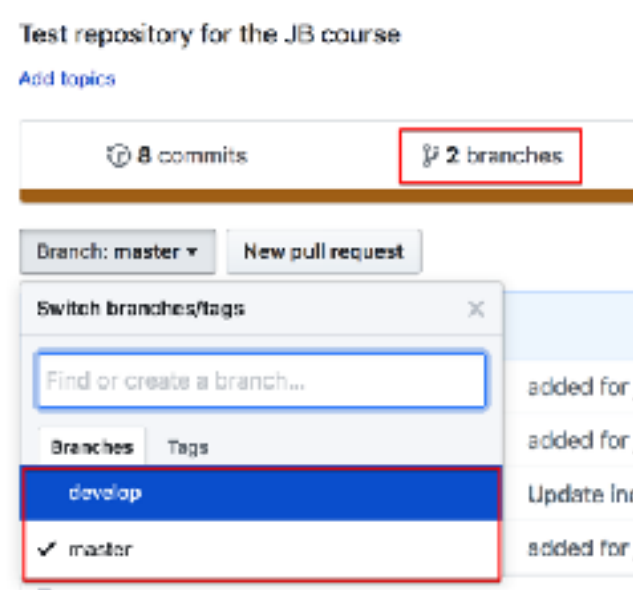
git branch

```
ubuntu@ubuntu-xenial:~/time_tracker/course$ git branch
* develop
  master
```

git push origin develop

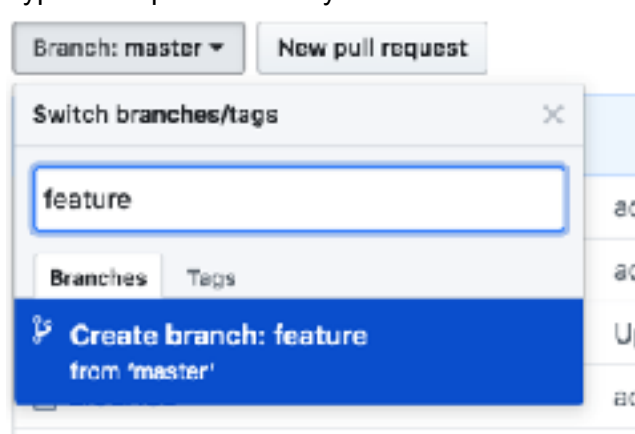
```
ubuntu@ubuntu-xenial:~/time_tracker/course$ git push origin develop
Username for 'https://github.com': [redacted]
Password for 'https://[redacted]:[redacted]@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com:[redacted]/time-tracker.git
 * [new branch]      develop -> develop
```

See the branch on the portal:



Create another branch “feature”, but this time using the GitHub web portal:

- On GitHub, navigate to the main page of the repository.
- Click the branch selector menu.
- Type a unique name for your new branch.

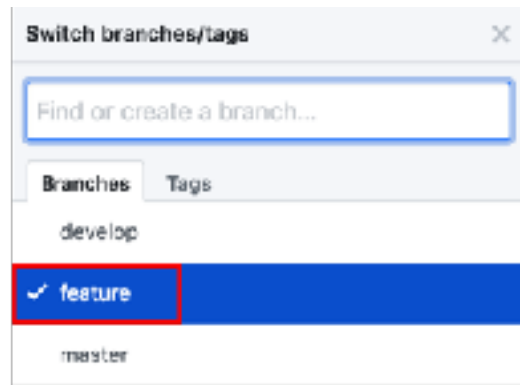


- Press **Enter**.

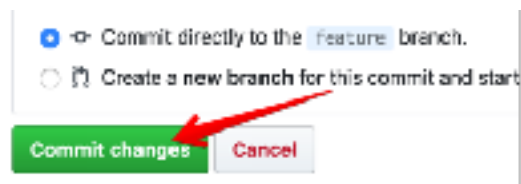
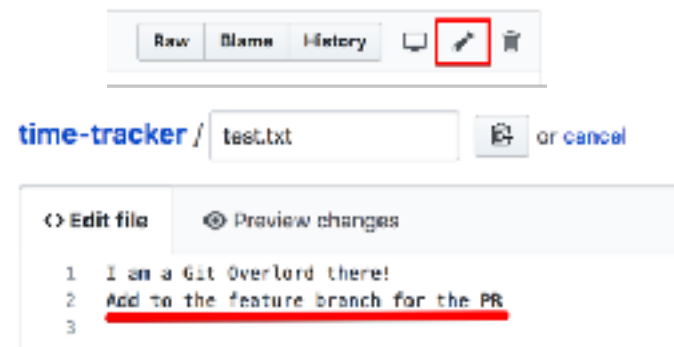
6. Create a Pull Request (PR)

To create a PR, we would use the GitHub web portal, however all aforementioned 3 branches are identical and we need to alter something before creating the PR:

Go to GitHub portal and switch to the feature branch:

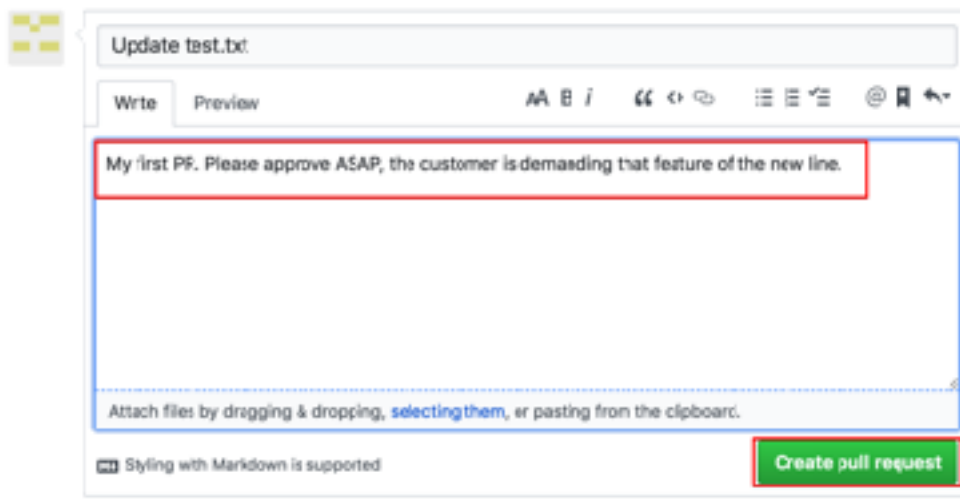


Find our test.txt and add a new line there:



Create a PR from 'feature' to 'develop'





Update test.txt:

Write Preview

My first PR. Please approve ASAP, the customer is demanding that feature of the new line.

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

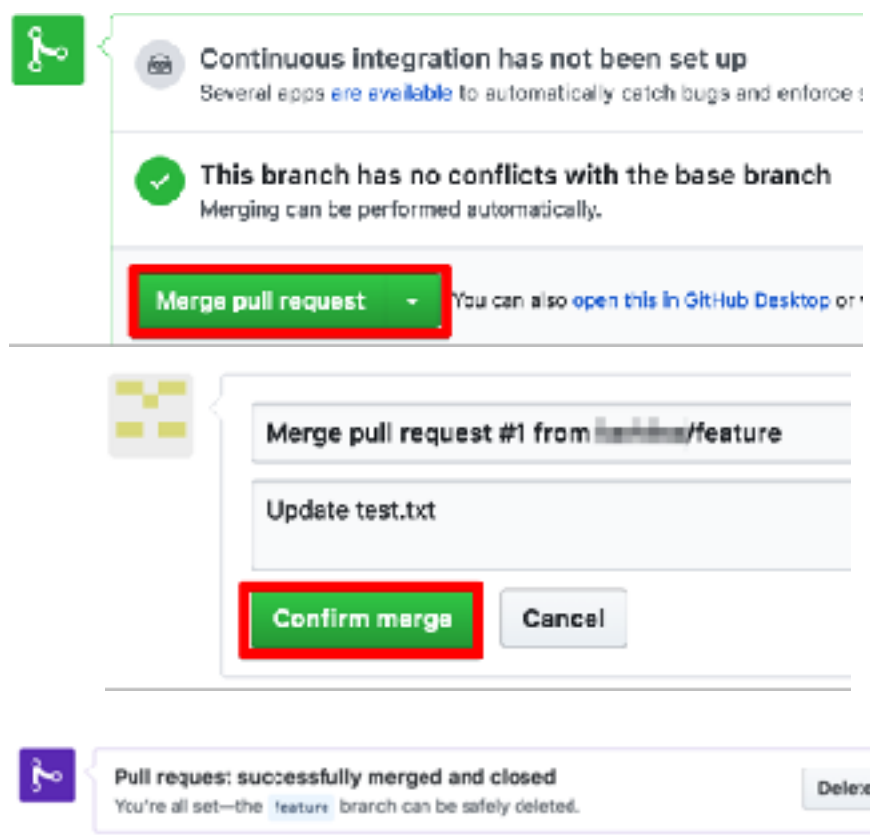
Styling with Markdown is supported

Create pull request

Go to Pull Requests:



Drill down our PR and Merge it:



Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce tests.

✓ 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 the command line.

Merge pull request #1 from [redacted]/feature

Update test.txt

Confirm merge Cancel

Pull request: successfully merged and closed
You're all set—the feature branch can be safely deleted.

Delete branch

Go to the develop branch and check the new line in the test.txt there