Web, Mobile and Security

Basic GIT flow for the labs: commands and their explanation

VERSION A (PREFERRED): This document applies only to those who have actually enabled SSH on op-gitlab.

0. Setting up SSH access on op-gitlab

Follow this procedure to do so: https://leho-howest.instructure.com/courses/7318/pages/git?module_item_id=107090

1. Creating a new project

1. Connect to your group via https://op-gitlab.howest.be/TI/2019-2020/s2-web-mobile-and-security/students/student-firstname-lastname
2. Create a new repository using the `New project` button.

2. Initializing an existing folder

1. In Windows Explorer, rightclick the folder you want to initialize and choose to open a GITBASH in that folder.
2. Next, make sure you initialize the folder as a GIT project by issuing the following command from within the GITBASH:

   ```
   git init
   ```

   This basically tells the system that the folder is from now on a GIT repository.
3. We will now define the "remote". The remote is the repo on the op-gitlab server to which we will push later on.

   ```
   git remote add origin git@op-gitlab.howest.be:TI/2019-2020/s2-web-mobile-and-security/students/student-firstname-lastname/01-lotto.git
   ```

   Make sure to replace `01-lotto.git` by the repo you created in section 1. You can also copy this line from the newly created repo:

4. Next, add the existing contents of the folder to the GIT system using:

```
git add .
```

5. You now need to commit these files locally, so they can be pushed remotely in a later stage. To do so, execute this command:

```
git commit -m "Initial commit"
```

   Of course, feel free to use another relevant commit message instead of `"Initial commit"`.

   More information on how to write commit messages will come later in the curriculum.

6. Finally, push the files towards the GIT server by issuing the following command:

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

3. Adding, committing and pushing new/modified files

1. From within the GITBASH, make sure to add the new/modified file(s):

```
git add .
```

2. Once happy, make sure to commit using an appropriate message, for example:

```
git commit -m "Add newfile.txt"
```

3. And finally do not forget to push:

```
git push
```