Document Code : **SP201902PGVA1**

GIT'S PRESENTATION

Changes History

Version	Date	Author	Description
1.0	08-02-2019	Eloundou Célestin P. Tema Gildas Fomena Landry	Initial version

Contents

INTRODUCTION

Every day in our live we use to have projects, and conduct them point to point until to their achievement. When conducing, we use to face some problems of version control, we want to track the evolution of our project, to know the current state of our project at each minute, day and period that is the reason why we have to use a version control tool. Git is very adapted for software development projects, and will be very helpful for us at ITS. In this document, we'll present to you people what is git, how to install it how to configure a project and some commands to version our project.

I- What is git?

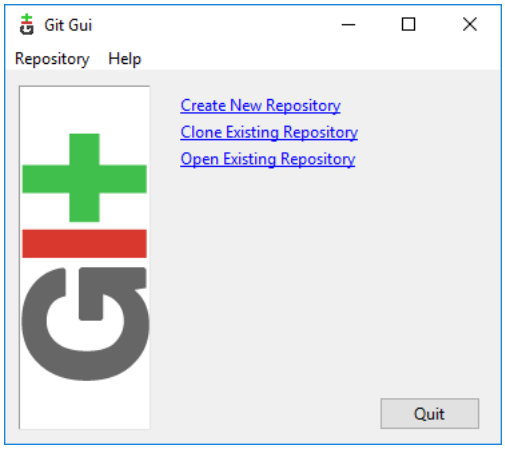
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. Git is based under many concepts as:

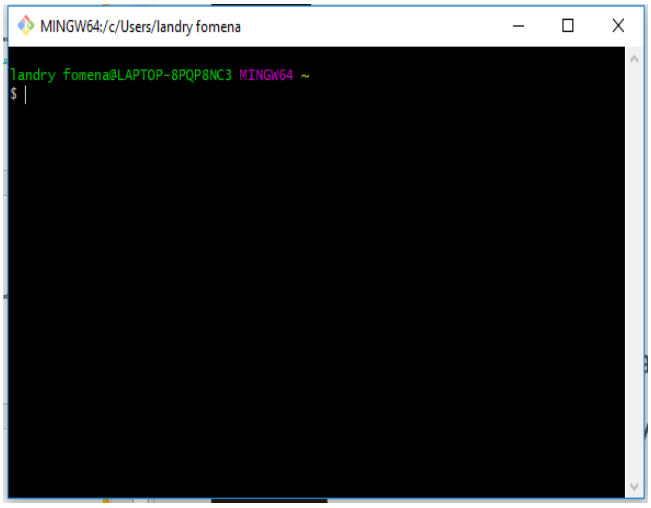
- **Branches**: you are able to create many branches of your project
- **Saving changes**: you can save your changes using commit commands
- **Pushing your changes**: you can send your changes on a remote repository using the push commands
- **Pulling the changes**: you can pull changes pushed by the other members of your project using the pull commands

II- How to install git?

Windows: git available here https://git-scm.com/downloads. Comes with git bash and git GUI

Mac and Linux: available in the system.





III- Some commands

To initiate git on a project: git *init*

```
landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git
$ git init
Initialized empty Git repository in C:/Users/landry fomena/Desktop/test git/.git
/
```

To create a new branch: git checkout -b nom_branch

```
landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (master)
$ git checkout -b nom_branch
Switched to a new branch 'nom_branch'

landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (nom_branch)
$
```

To add changes :

1rst step: git status

```
landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (nom_branch)
$ git status
On branch nom_branch

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        "Presentation de la consomation de l\342\200\231api des sms.pptx"
        Soutenance Stage BB.pptx
        pom.xml

nothing added to commit but untracked files present (use "git add" to track)

landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (nom_branch)
$
```

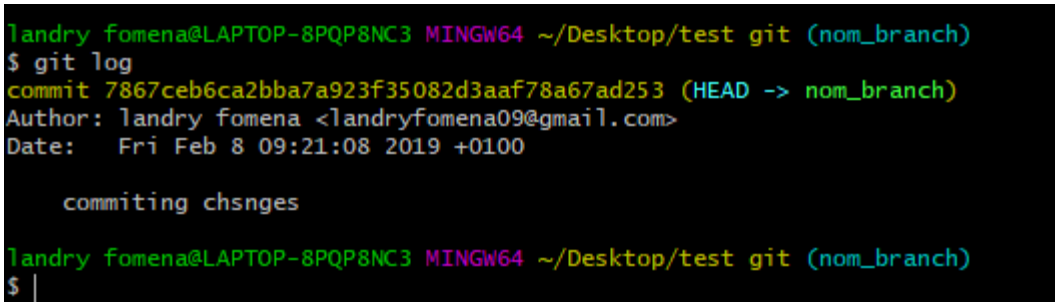
2nd step git add .

```
landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (nom_branch)
$ git add .

landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (nom_branch)
$ git status
On branch nom_branch

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   "Presentation de la consomation de l\342\200\231api des sms.
pptx"
        new file:   Soutenance Stage BB.pptx
        new file:   pom.xml


landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (nom_branch)
$
```

3rd step commiting changes :

```
landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (nom_branch)
$ git commit -m "commiting chsnges"
[nom_branch (root-commit) 7867ceb] commiting chsnges
 3 files changed, 61 insertions(+)
 create mode 100644 "Presentation de la consomation de l\342\200\231api des sms.
pptx"
 create mode 100644 Soutenance Stage BB.pptx
 create mode 100644 pom.xml

landry fomena@LAPTOP-8PQP8NC3 MINGW64 ~/Desktop/test git (nom_branch)
$
```

To see everything you did : git log



To see the full documentation of git you can follow this link: https://git-scm.com/doc

Conclusion

In this document we saw how to install git and use it for the versioning of our projects. Now feel free to use it as you want and stop worrying about versioning.