

Git et Github



Plan

- Présentation
- Les besoins
- Les notions clés
- Installation et commandes
- Les révisions
- Les branches
- GitHub

Présentation

Git est un logiciel de contrôle de version décentralisé et distribué.

Git a été inventé en 2005 par Linus Torvalds.

Git succède à de nombreux outils similaires comme cvs ou subversion.

Le principal concurrent de git est **Mercurial**

Les besoins



INTI
FORMATION

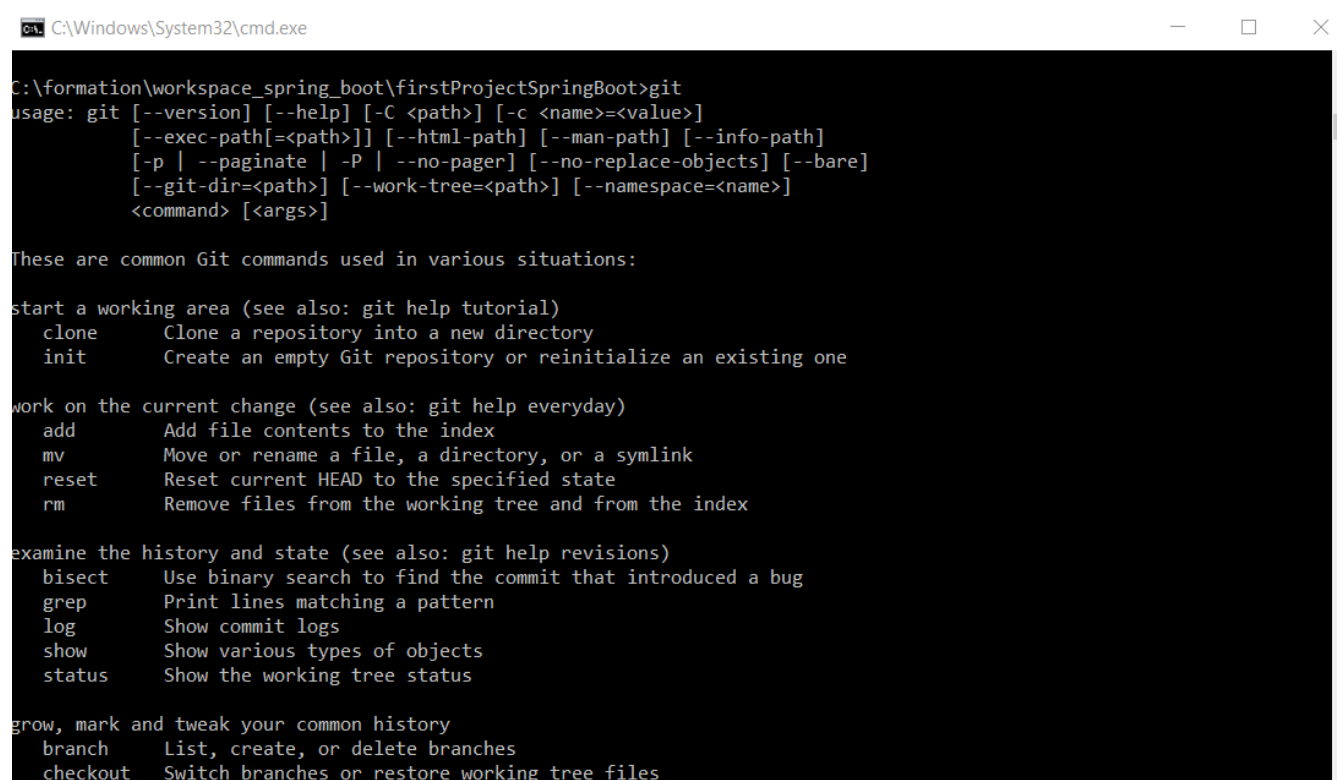
- Suivre les étapes de modification d'un programme.
- Tester un changement complexe et pouvoir revenir facilement en arrière.
- Travailler à plusieurs sur un projet.
- Inviter des collaborateurs sur un projet

Les notions clés



INTI
FORMATION

Ligne de Commande : Le programme de l'ordinateur que nous utilisons pour entrer des commandes Git. C'est un programme non-natif que vous téléchargez lorsque vous téléchargez Git pour la première fois. Vous tapez à l'écran des commandes à base de texte, appelées invites de commande, au lieu d'utiliser une souris.



```
C:\Windows\System32\cmd.exe

C:\formation\workspace_spring_boot\firstProjectSpringBoot>git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
    clone      Clone a repository into a new directory
    init       Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
    add        Add file contents to the index
    mv         Move or rename a file, a directory, or a symlink
    reset      Reset current HEAD to the specified state
    rm         Remove files from the working tree and from the index

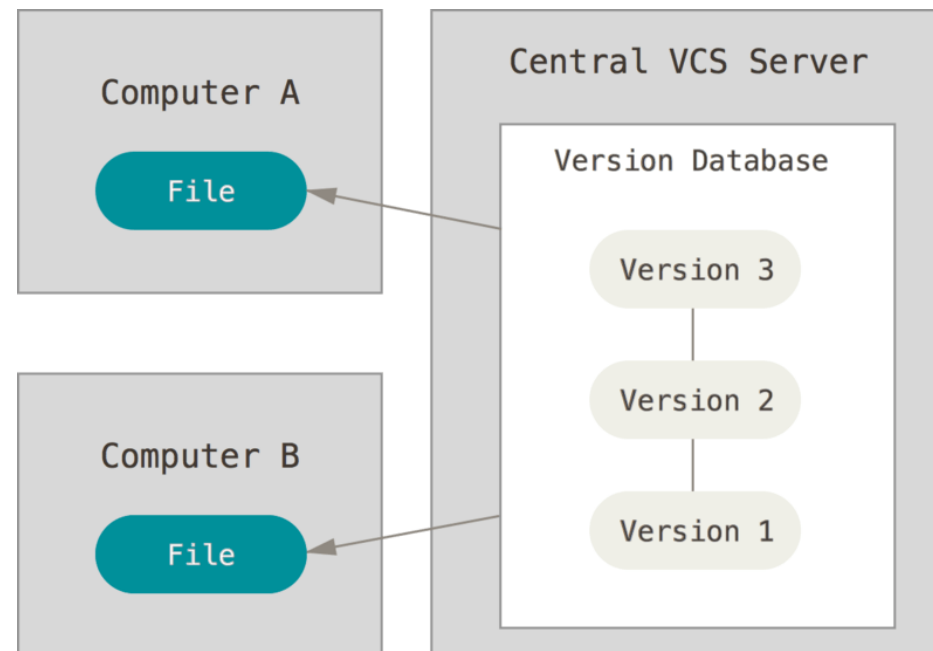

examine the history and state (see also: git help revisions)
    bisect     Use binary search to find the commit that introduced a bug
    grep       Print lines matching a pattern
    log        Show commit logs
    show       Show various types of objects
    status     Show the working tree status


grow, mark and tweak your common history
    branch     List, create, or delete branches
    checkout   Switch branches or restore working tree files
```


Dépôt : Un répertoire ou de l'espace de stockage où vos projets peuvent vivre. Parfois les utilisateurs GitHub raccourcissent ça en « repo ». Il peut être local sur un répertoire de votre ordinateur, ou ce peut être un espace de stockage sur GitHub ou un autre hébergeur en ligne. À l'intérieur d'un dépôt, Vous pouvez conserver des fichiers de code, des fichiers texte, des images.



Contrôle de Version : Fondamentalement, l'objectif pour lequel Git a été conçu. Quand vous avez un fichier Microsoft Word, vous l'écrasez à chaque fois que vous faites une nouvelle sauvegarde, ou vous sauvegardez plusieurs versions. Avec Git, vous n'êtes plus obligé de faire ça. Git conserve des « instantanés » de chaque point dans l'historique d'un projet, de sorte que vous ne pouvez jamais le perdre ou l'écraser.



Commit : C'est la commande qui donne à Git toute sa puissance. Quand vous faites un « commit », vous prenez un « instantané », une « photo » de votre dépôt à ce stade, vous donnant un point de contrôle que vous pouvez ensuite réévaluer ou restaurer votre projet à un état précédent.

Branche : Comment faire travailler plusieurs personnes sur un projet en même temps sans que Git ne s'embrouille ? Habituellement, elles se « débranchent » du projet principal avec leurs propres versions complètes des modifications qu'elles ont chacune produites de leur côté. Après avoir terminé, il est temps de « fusionner » cette branche pour la ramener vers la branche « master », le répertoire principal du projet.

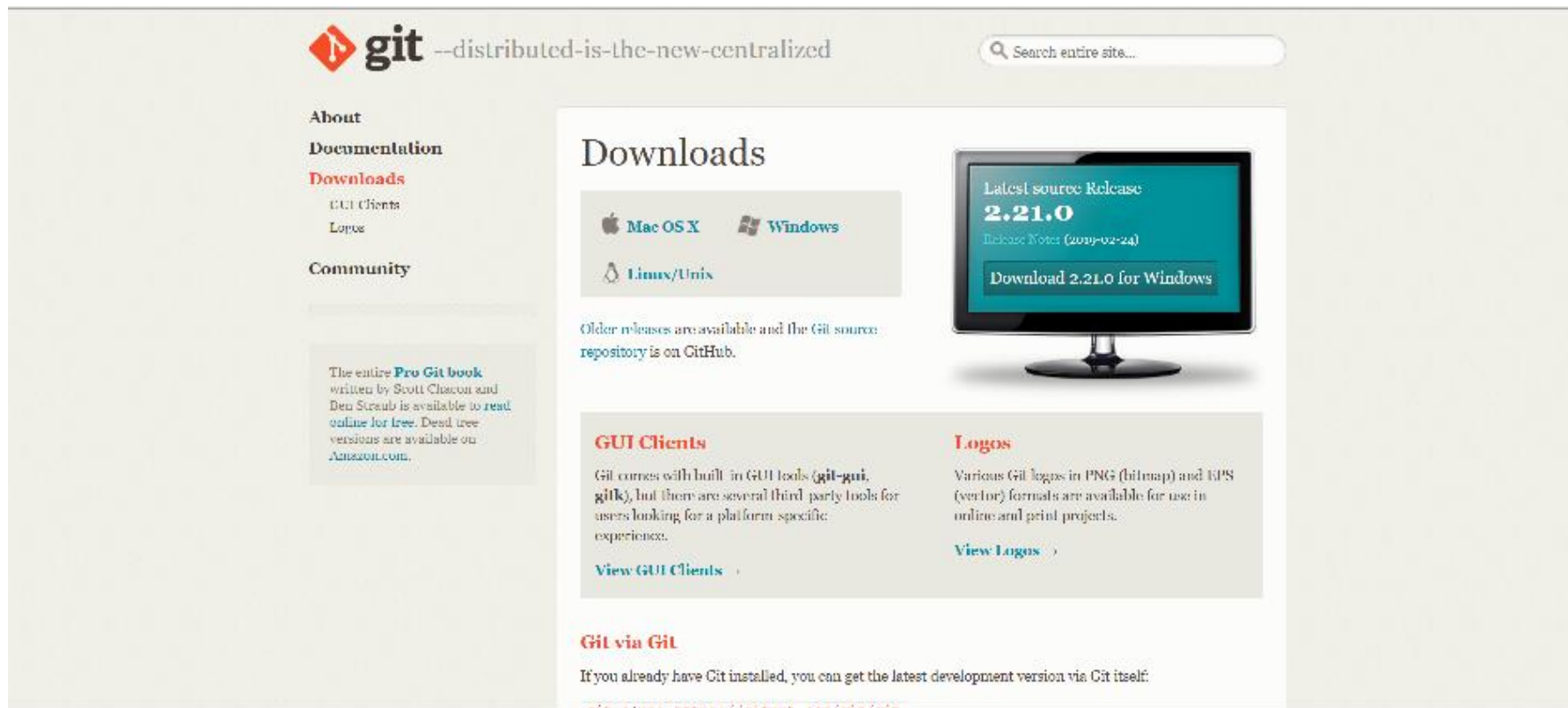
Installation et commandes



INTI
FORMATION

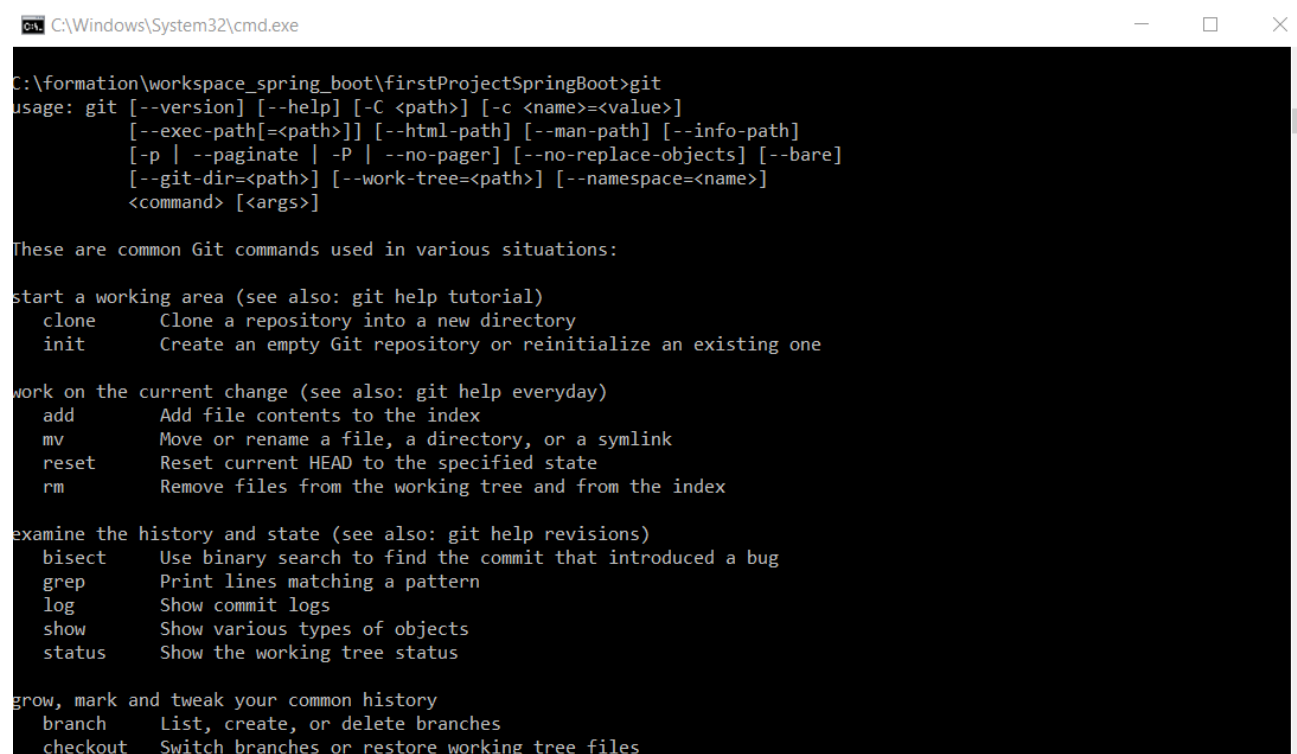
Installer git ?

- Téléchargez et installez la dernière version de **Git** :
<https://git-scm.com/downloads>



Comment utiliser git ?

- **Git** est un outil fonctionnant en ligne de commande.
- La commande s'appelle git.
- Elle est utilisable depuis un terminal.



```
C:\Windows\System32\cmd.exe

C:\formation\workspace_spring_boot\firstProjectSpringBoot>git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
    clone      Clone a repository into a new directory
    init       Create an empty Git repository or reinitialize an existing one

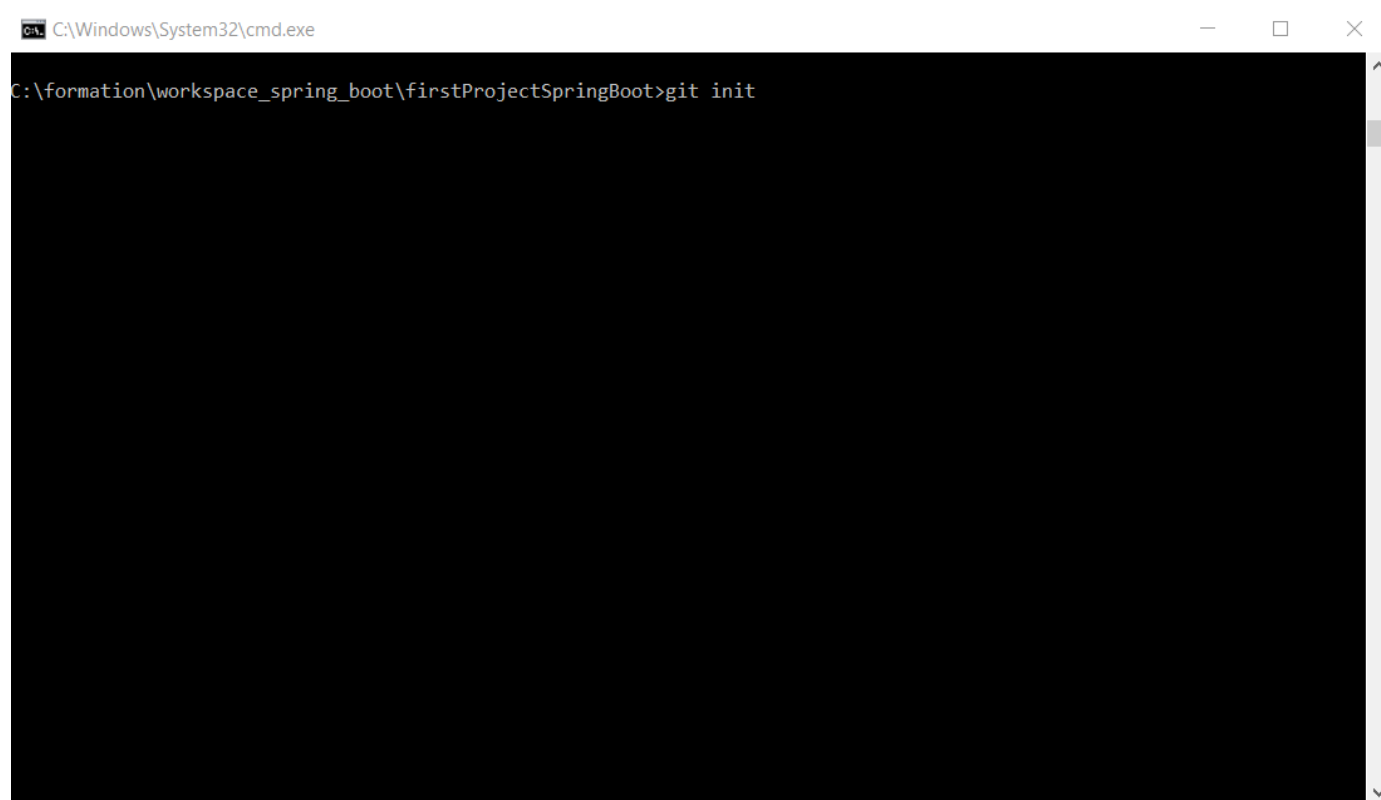

work on the current change (see also: git help everyday)
    add        Add file contents to the index
    mv         Move or rename a file, a directory, or a symlink
    reset      Reset current HEAD to the specified state
    rm         Remove files from the working tree and from the index


examine the history and state (see also: git help revisions)
    bisect     Use binary search to find the commit that introduced a bug
    grep       Print lines matching a pattern
    log        Show commit logs
    show       Show various types of objects
    status     Show the working tree status


grow, mark and tweak your common history
    branch     List, create, or delete branches
    checkout   Switch branches or restore working tree files
```

Les commandes :

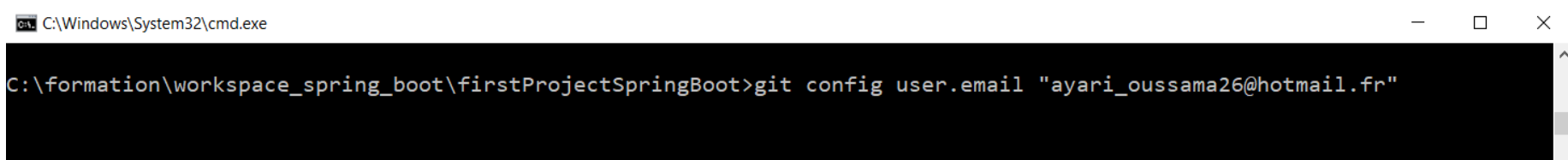
> **git init** : Initialise un nouveau dépôt Git. Jusqu'à ce que vous exécutiez cette commande dans un dépôt ou répertoire, c'est juste un dossier ordinaire. Seulement après avoir entré cette commande, il accepte les commandes Git qui suivent.



```
C:\Windows\System32\cmd.exe
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git init
```

Les commandes :

> **git config** : raccourci de "configurer", ceci est tout particulièrement utile quand vous paramétrez Git pour la première fois.

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Windows\System32\cmd.exe". The command prompt shows the current directory as "C:\formation\workspace_spring_boot\firstProjectSpringBoot" and the command "git config user.email 'ayari_oussama26@hotmail.fr'" being entered. The command prompt has a black background with white text. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
C:\Windows\System32\cmd.exe  
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git config user.email "ayari_oussama26@hotmail.fr"
```


Les commandes :

> **git status** : Vérifie le statut de votre repository. Voir quels fichiers sont à l'intérieur, quelles sont les modifications à commuter, et sur quelle branche du repository vous êtes en train de travailler.

```
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git status
On branch master

No commits yet

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

        .gitignore
        .mvn/
        mvnw
        mvnw.cmd
        pom.xml
        src/

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

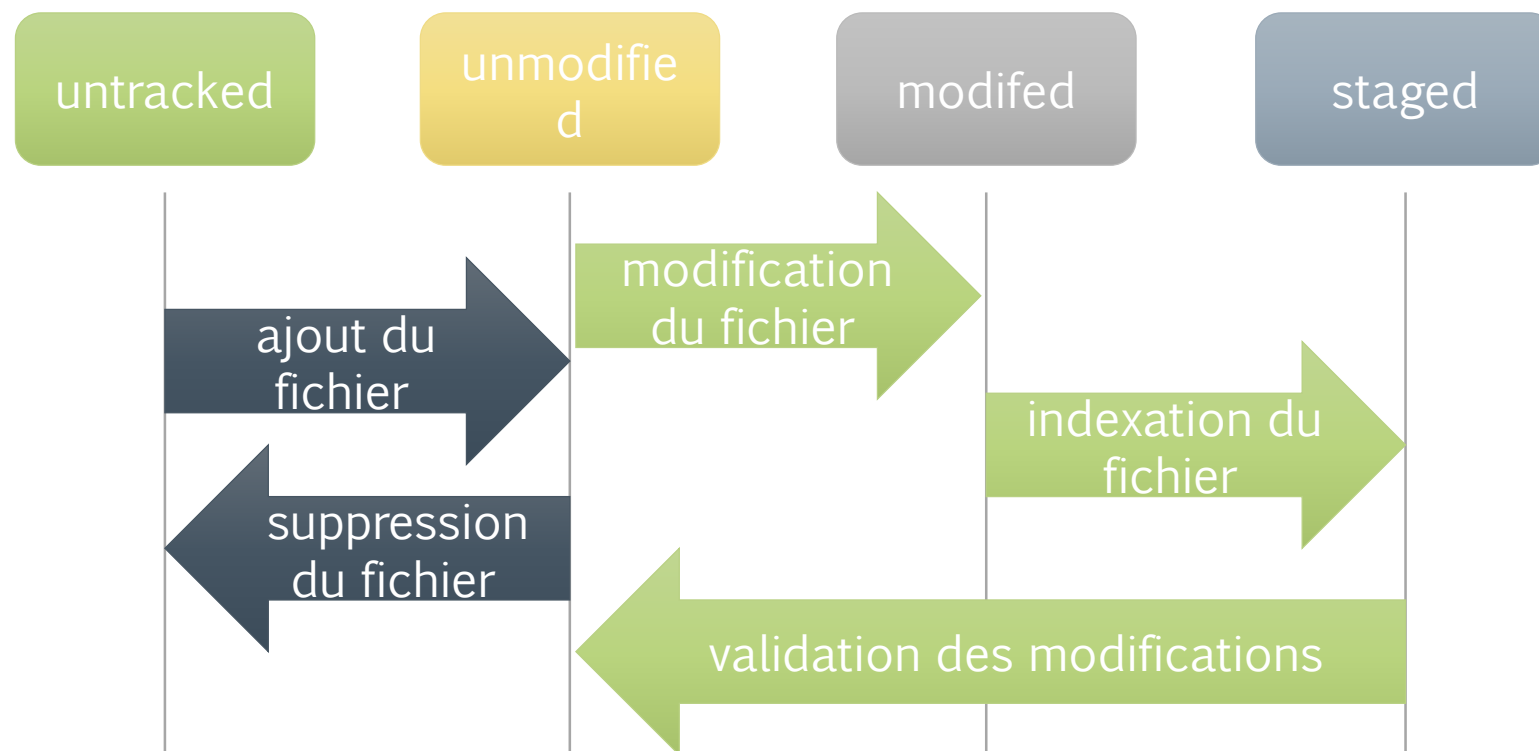
Les commandes :

> **git status** : On retrouve 4 états possibles pour chaque fichier :

- untracked : fichier non suivi
- unmodified : fichier non modifié
- modified : fichier modifié
- staged : modification du fichier indexée

Les commandes :

> git status



Les commandes :

➤ **git add** : Ceci n'ajoute pas de nouveaux fichiers dans votre repository. Au lieu de cela, ceci porte de nouveaux fichiers à l'attention de Git. Après avoir ajouté des fichiers, ils sont inclus dans les « instantanés » du dépôt Git.

```
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git add -A
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .mvn/wrapper/MavenWrapperDownloader.java.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .mvn/wrapper/maven-wrapper.properties.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in mvnw.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in mvnw.cmd.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in pom.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main/java/com/adaming/FirstProjectSpringBootApplication.java.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/test/java/com/adaming/firstProjectSpringBoot/FirstProjectSpringBootApplicationTests.java.
The file will have its original line endings in your working directory
```

Les commandes :

- **git commit** : la commande la plus importante de Git. Après avoir effectué toute sorte de modification, vous entrez ça afin de prendre un "instantané" du dépôt. Généralement cela s'écrit sous la forme : **git commit -m "Message ici"** Le -m indique que la section suivante de la commande devrait être lue comme un message.

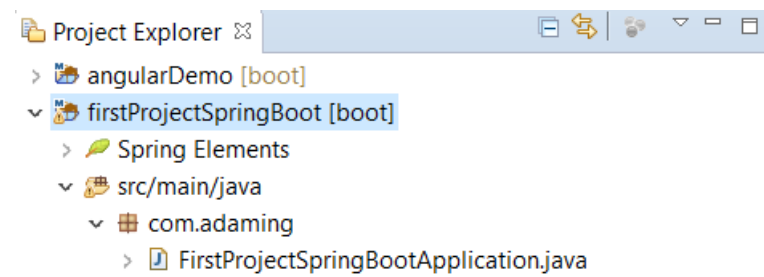
```
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git commit -m "add all files"
[master (root-commit) b03a337] add all files
28 files changed, 1440 insertions(+)
create mode 100644 .gitignore
create mode 100644 .mvn/wrapper/MavenWrapperDownloader.java
create mode 100644 .mvn/wrapper/maven-wrapper.jar
create mode 100644 .mvn/wrapper/maven-wrapper.properties
create mode 100644 mvnw
create mode 100644 mvnw.cmd
create mode 100644 pom.xml
create mode 100644 src/main/java/META-INF/MANIFEST.MF
create mode 100644 src/main/java/com/adaming/FirstProjectSpringBootApplication.java
create mode 100644 src/main/java/com/adaming/config/JWTAuthenticationFilter.java
create mode 100644 src/main/java/com/adaming/config/JWTAuthorizationFilter.java
create mode 100644 src/main/java/com/adaming/config/SecurityConfig.java
create mode 100644 src/main/java/com/adaming/config/SecurityConstants.java
create mode 100644 src/main/java/com/adaming/controller/LoginController.java
create mode 100644 src/main/java/com/adaming/controller/RoleController.java
create mode 100644 src/main/java/com/adaming/controller/UtilisateurController.java
create mode 100644 src/main/java/com/adaming/entities/Role.java
create mode 100644 src/main/java/com/adaming/entities/Utilisateur.java
create mode 100644 src/main/java/com/adaming/model/UtilisateurDetail.java
create mode 100644 src/main/java/com/adaming/repositories/RoleRepository.java
create mode 100644 src/main/java/com/adaming/repositories/UtilisateurRepository.java
create mode 100644 src/main/java/com/adaming/services/AppUserDetailsService.java
create mode 100644 src/main/java/com/adaming/services/IRoleService.java
create mode 100644 src/main/java/com/adaming/services/IUtilisateurService.java
create mode 100644 src/main/java/com/adaming/services/RoleService.java
create mode 100644 src/main/java/com/adaming/services/UtilisateurService.java
create mode 100644 src/main/resources/application.properties
```

Ajouter les fichiers à son dépôt ?

Mon dépôt git



Mon projet



Introduisons une erreur : Modifiez le fichier Utilisateur.java afin d'y introduire un mauvais syntaxe pour l'attribut nom.

```
private static final long serialVersionUID = 1L;
```

```
@Id
```

```
@GeneratedValue(strategy = GenerationType.AUTO)
```

```
private Long idUtilisateur;
```

```
private String nomm;
```

```
private String prenom;
```

```
@Column(unique = true)
```

```
private String username;
```

```
private String password;
```

```
@Lob
```

Retrouver la version précédente d'un fichier : Restaurer la dernière version valide du fichier à l'aide de la commande git checkout

```
C:\formation\workspace_spring_boot\firstProjectSpringBoot\src\main\java\com\adaming\entities>git checkout Utilisateur.java
```

```
Utilisateur.java
17 @Entity
18 public class Utilisateur implements Serializable {
19     /**
20      *
21      */
22     private static final long serialVersionUID = 1L;
23
24     @Id
25     @GeneratedValue(strategy = GenerationType.AUTO)
26     private Long idUtilisateur;
27
28     private String nom;
29
30     private String prenom;
31
32     @Column(unique = true)
```

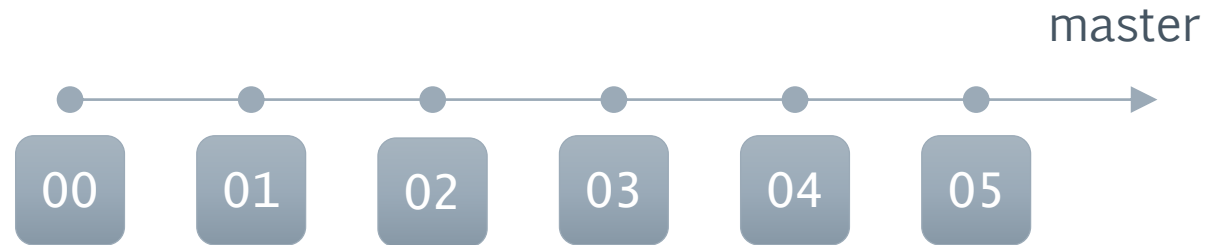

Les révisions



INTI
FORMATION

Chaque commit sur votre dépôt crée une nouvelle révision ou version de votre projet.

Chaque révision correspond en fait à un ensemble de changement que vous avez « comité »



Pour consulter les révisions dans notre dépôt local, on utilise la commande :

```
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git log
commit b03a33717e154155612869c73df30fd0671d4492 (HEAD -> master, origin/master)
Author: Oussama AYARI <ayari_oussama26@hotmail.fr>
Date: Thu May 9 05:11:58 2019 +0200

    add all files

C:\formation\workspace_spring_boot\firstProjectSpringBoot>
```

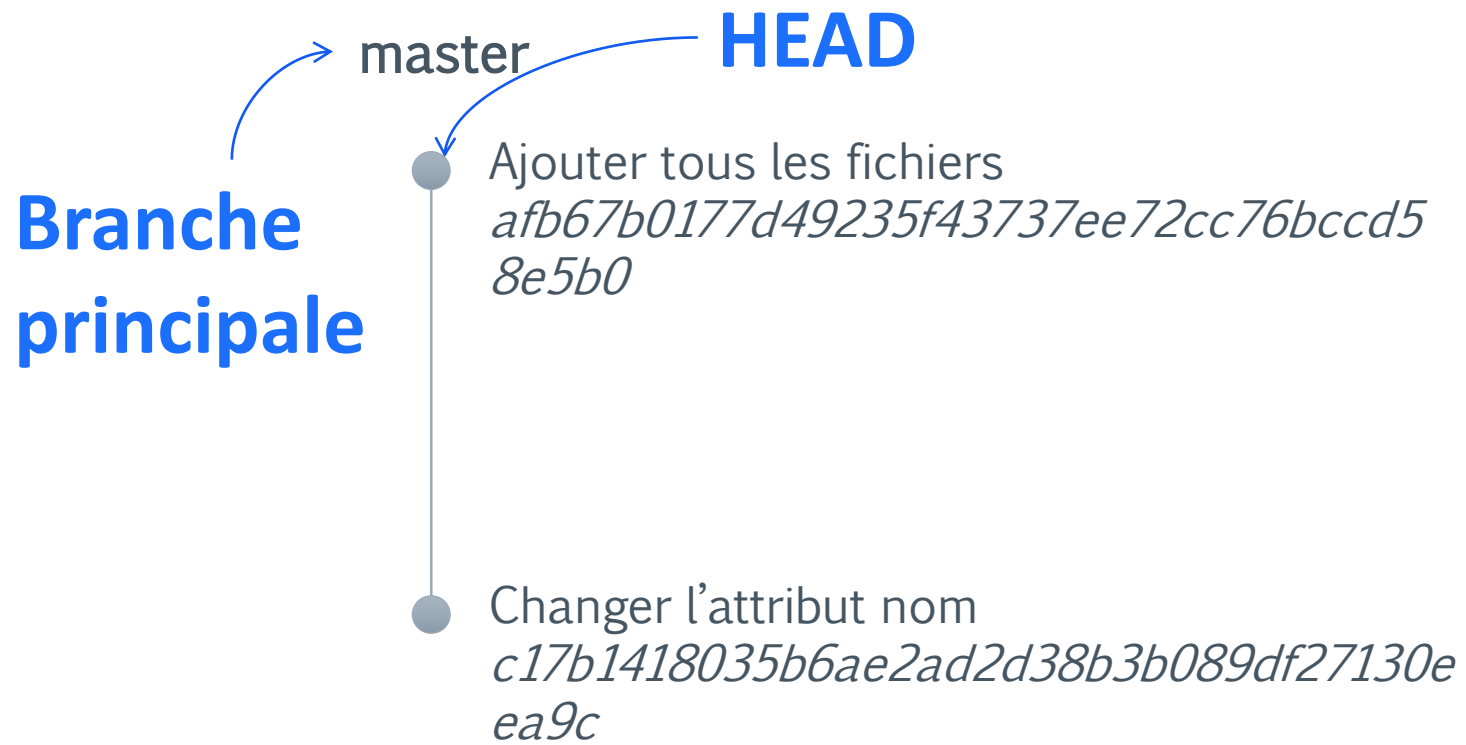
Pour chaque révision, on peut consulter les informations suivantes :

- **commit** : numéro de révision
- **Author** : utilisateur ayant enregistré la révision
- **Date** : date de la création de la révision
- **Commentaire** inscrit par l'utilisateur lors du commit

Les branches



INTI
FORMATION



master



**DETACHED
HEAD**

Changer l'attribut nom

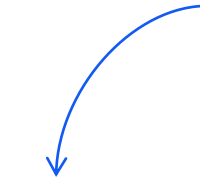
c17b1418035b6ae2ad2d38b3b089df27130eea9c

**git checkout
c17b14...**

master

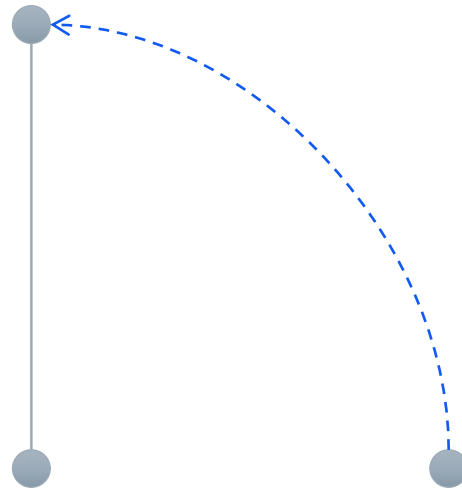


dev



**On peut
démarrer une
nouvelle
branche**

master



**On peut
revenir à la
version
courante
git checkout
master**

Créer une nouvelle branche :

```
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git branch local
```

Consultez les branches disponibles sur votre dépôt :

```
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git branch  
local  
* master
```

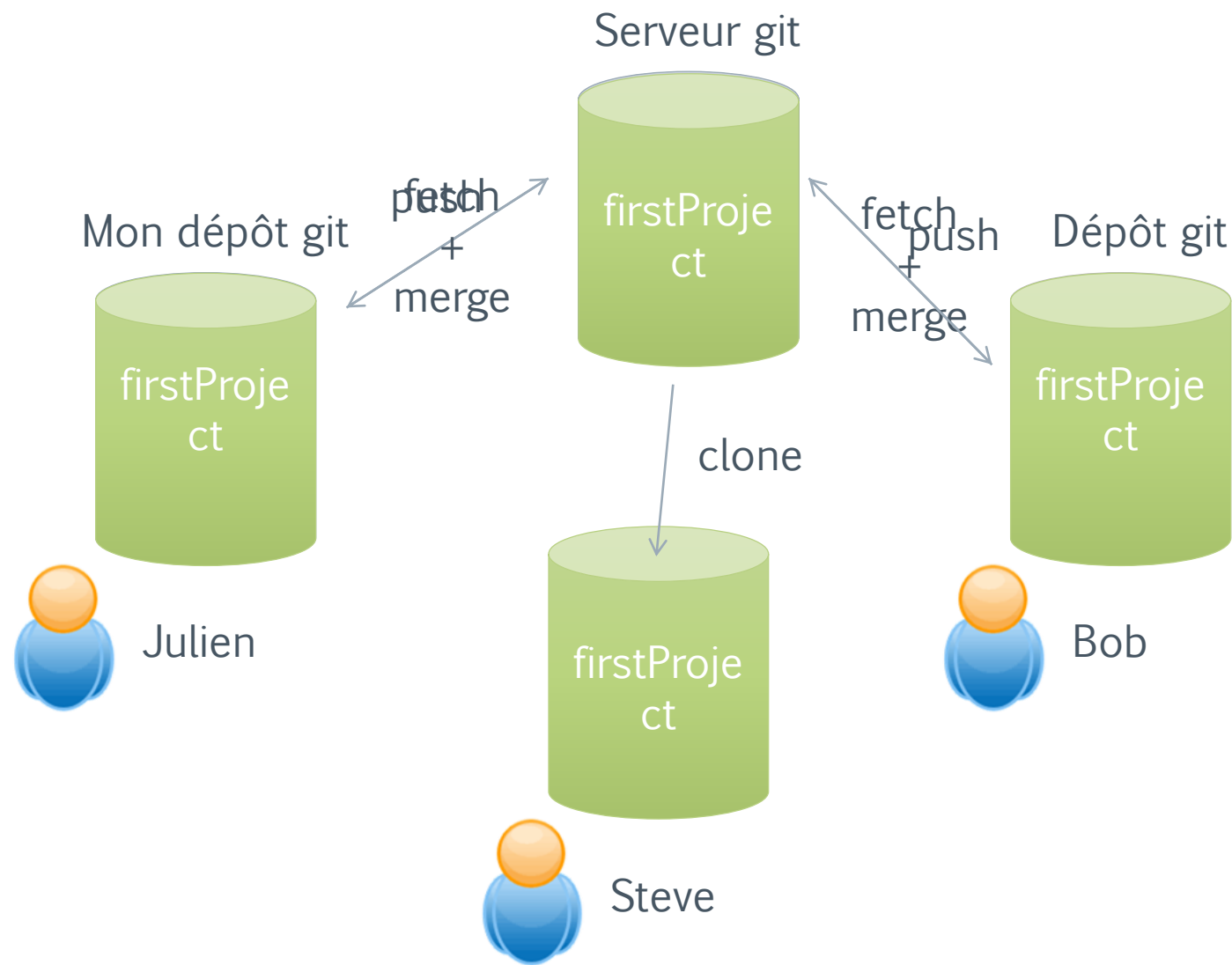
Changer de branche :

```
C:\formation\workspace_spring_boot\firstProjectSpringBoot>git checkout local  
Switched to branch 'local'
```

GitHub



INTI
FORMATION



Le cycle git

