

# GIT - INTRODUCTION

---

Git est un logiciel de gestion de versions (créé par Linus Torvald à l'origine du noyau Linux).

Un logiciel de gestion de versions (en Français logiciel de versioning) est un outil utilisé pour gérer les codes sources d'un projet informatique et propose en général les fonctionnalités de base suivantes:

- Suivi de l'évolution d'un fichier (ajout, modification, déplacement, suppression)
- Suivi du travail des développeurs
- Travail collaboratif sur un même fichier
- Gestion de différentes versions d'un projet en simultané

## GIT - Installation

---

Debian (directement à partir du terminal):

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

Windows:

- <https://git-scm.com/downloads>

Pour la suite de ce cours:

Toutes les commandes présentées à la suite de ce cours seront à lancer dans le terminal:

- le terminal par défaut pour les ordinateurs sous Linux
- avec Git Bash pour les ordinateurs sous Windows

# GIT - Configuration

---

Renseigner votre identité de développeur

```
git config --global user.name "Votre nom"  
git config --global user.email nom@domaine.fr
```

Activer les couleurs automatiques (pour plus de lisibilité)

```
git config --global color.diff auto  
git config --global color.status auto  
git config --global color.branch auto
```

## GIT - Les dépôts

---

Un dépôt représente une copie d'un projet informatique. Tous les développeurs travaillant sur un projet possèdent une copie du dépôt.

Chaque dépôt contient les fichiers du projet ainsi que tout l'historique des modifications depuis sa création.

## GIT - Créer un nouveau dépôt

Lancer le terminal Créer un répertoire du nom du projet à créer sur le disque local.

Exemple :

```
mkdir mon1erProjet  
cd mon1erProjet
```

Une fois dans le répertoire, nous pouvons initialiser le nouveau dépôt:

```
git init
```

Cette commande aura pour effet de créer un répertoire `.git` dans le dossier de votre projet. Ce répertoire `.git` contient la configuration de votre dépôt ainsi que tout l'historique de tous les fichiers et branches du projet.

Une fois cette commande entrée, le nouveau projet GIT est maintenant créé et prêt à être utilisé.

## GIT - Cloner un dépôt existant

Cloner un dépôt consiste à récupérer une nouvelle copie d'un projet.

Il est possible de cloner n'importe quel dépôt public existant (par exemple sur GitHub) ou tout dépôt privé auquel vous pourriez avoir accès.

Par exemple, pour cloner le projet Symfony bien connu des développeurs Web, il suffit d'entrer la commande suivante:

```
git clone http://github.com/symfony/symfony.git
```

Git télécharge une copie du projet et l'enregistre dans un répertoire dont le nom correspond au nom du projet cloné (ici symfony).

```
git clone http://github.com/symfony/symfony.git
Initialized empty Git repository in /home/git/symfony/.git/
remote: Counting objects: 7891, done.
remote: Compressing objects: 100% (2510/2510), done.
remote: Total 7891 (delta 5117), reused 7982 (delta 6552)
Receiving objects: 100% (7891/7891), 1.42 MiB | 688 KiB/s, done.
Resolving deltas: 100% (5117/5117), done.
Checking out files: 100% (571/571), done.
```

Note:

Pour créer un dépôt qui servira de serveur, ajouter le paramètre `--bare` à `git init` ou `git clone`.

Ce paramètre créera/clonera un projet qui ne contiendra que l'historique des modifications (pas d'édition de code source sur cette copie du projet).

## GIT – Le fichier de configuration

Chaque dépôt possède un fichier de configuration unique. Il est localisé dans le répertoire `.git` du projet.

Voici à quoi ressemble ce fichier de configuration :

```
[core]

    repositoryformatversion = 0

    filemode = false

    bare = false

    logallrefupdates = true

    symlinks = false

    ignorecase = true

[remote "origin"]

    url = ssh://git@mondomaine.fr/home/git/Mon1erDepot

    fetch = +refs/heads/*:refs/remotes/origin/*

[branch "master"]

    remote = origin

    merge = refs/heads/master
```

Les 2 sections qui nous intéressent sont `[remote "origin"]` et `[branch "master"]`

La section `[remote "origin"]` contient les informations du serveur distant. « origin » correspond au nom que l'on a donné au serveur distant (origin étant le nom par défaut) afin de pouvoir y accéder plus facilement à l'exécution de certaines commandes. L'information importante dans cette section est le paramètre `url` qui correspond à l'adresse du dépôt distant.

Dans l'exemple ci-dessus, le dépôt distant `origin` est situé à `ssh://git@mondomaine.fr/home/git/Mon1erDepot`

## GIT - Utilisation d'un dépôt local

Lancez le terminal et dirigez vous vers le répertoire d'un projet GIT existant (créé par vous même ou cloné comme vu ci-dessus).

Voici la liste des commandes de base pour utiliser votre dépôt:

<code>git status</code> Indique les récentes modifications effectuées sur le dépôt.	Aucune modification: # On branch master nothing to commit (working directory clean)  Sinon, liste des modifications
<code>git add chemin/vers/le/fichier</code> ajoute un fichier au dépôt	Pour ajouter tous les fichiers: <code>git add ./*</code>  Pour mettre à jour l'index: <code>git add -u</code>  Pour mettre à jour tous les fichiers: <code>git add -a</code>
<code>git commit -m "raison du commit"</code> Sauvegarde les modifications en local	
<code>git push origin master</code> Pousse les commits vers un serveur distant	<code>origin</code> correspond au nom du serveur enregistré. <code>master</code> correspond au nom de la branche.

## GIT – Synchronisation avec un serveur distant

Un dépôt git peut-être stocké sur un serveur. Avant toute modification, il est indispensable de récupérer les modifications éventuellement faites par d'autres développeurs ou par vous-même sur une autre machine.

Pour récupérer les dernières modifications du projet à partir du dépôt distant :

```
git pull origin master
```

`origin` correspond au serveur où est stocké le dépôt distant (voir section « GIT – le fichier de configuration »)

`master` correspond à la branche de travail (les branches ne seront pas traitées ici)

Une fois vos modifications enregistrées (via la commande `commit`, voir GIT - Utilisation d'un dépôt local), il est temps de les « pousser » (sauvegarder) sur le serveur distant.

La commande à entrer est la suivante :

```
git push origin master
```

Vos modifications seront alors poussées vers le serveur. Les autres développeurs peuvent désormais exécuter la commande `git pull...` sur leur poste afin de récupérer vos modifications.

## GIT – Conclusion

Nous avons vu ici comment initialiser, utiliser et sauvegarder un dépôt GIT pour un projet informatique.

Pour aller plus loin, vous pouvez consulter le fichier PDF « Git Livre.pdf » disponible dans le même répertoire que le document courant.