GIT

|  |  |  |
| --- | --- | --- |
| Schatz Johanne | 12 avril 2023 | Simplon formations |

# SOMMAIRE

[GIT 1](#_Toc1285702210)

[SOMMAIRE 1](#_Toc1066356107)

[I. Théorie 2](#_Toc901473959)

[La gestion de versions 2](#_Toc2133643208)

[A quoi sert la gestion de versions dans un projet de développement informatique ? 2](#_Toc507504364)

[Qu'est ce que Git ? 2](#_Toc1190864992)

[La différence entre Git et GitHub 3](#_Toc472454432)

[GitHub 3](#_Toc54301563)

[Quelques plateformes d’hébergement de code 3](#_Toc1119808834)

[GitHub 3](#_Toc1783217231)

[GitLab 3](#_Toc133845990)

[Bitbucket 4](#_Toc1538703227)

[Historique de Git 4](#_Toc55746884)

[Les particularités de Git Git Workflow 4](#_Toc453161869)

[II. Pratique 5](#_Toc522391722)

[[Soignez votre présentation] 5](#_Toc916946914)

[[Titre 3] 5](#_Toc1091456163)

# Théorie

## La gestion de versions

Un gestionnaire de versions est un programme qui permet aux développeurs de conserver un historique des modifications et des versions de tous leurs fichiers.

Il garde en mémoire :

* Chaque modification de chaque fichier
* Pourquoi elle a eu lieu
* Qui l’a faite

## A quoi sert la gestion de versions dans un projet de développement informatique ?

3 grandes fonctionnalités

* Revenir à une version précédente de votre code en cas de problème
* Suivre l’évolution de votre code étape par étape
* Travailler à plusieurs sans risque de supprimer les modifications des autres collaborateurs

Qu'est ce que Git ?

Git est un programme informatique. C’est un gestionnaire de versions avec une structure décentralisée : c’est à dire que chaque copie du code effectué correspond à un nouveau dépôt dans lequel est conservé l’historique des modifications. L’historique complet du code n’est pas conservé dans un unique emplacement.

Un dépôt = a repository en anglais.

## La différence entre Git et GitHub

Git est utilisé pour créer un dépôt local et gérer les versions de vos fichiers alors que GitHub est un service en ligne qui va héberger votre dépôt, on parle dans ce cas de dépôt distant.

Avec Git vous préparez votre code, avec GitHub vous stockez votre code et pouvez le partager.

## GitHub

Git et GitHub (ou un autre service d’hébergement de versions de code) fonctionnent donc ensemble. Le dépôt local est un clone du dépôt distant. Je fais toutes les modifications de mon code dans ma copie locale et puis j’envoie ma nouvelle version vers le dépôt distant pour que le code du dépôt distant soit mis à jour.

Les dépôts sont utiles si :

* Vous souhaitez conserver un historique de votre projet.
* Vous travaillez à plusieurs.
* Vous souhaitez collaborer à des projets open sources.
* Vous devez retrouvez par qui a été faite chaque modification
* Vous voulez savoir pourquoi chaque modification a eu lieu.

## Quelques plateformes d’hébergement de code

### GitHub

Outil de communication et de collaboration entre plusieurs développeurs (ou toute personne qui écrit du texte), son interface web a été créée pour faciliter l’interaction avec Git.

Il est devenu le portfolio des développeurs et permet de mettre en avant la qualité de son code.

### GitLab

Principale alternative à GitHub depuis le rachat de GitHub par Microsoft. Il a des outils de CI CD intégrés et est gratuit pour les petites structures.

### Bitbucket

Version d’hébergeur de code de la société Atlassian.

Source : OpenClassrooms Gérez du code avec Git et GitHub

## Historique de Git

**Git** est un [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) [décentralisé](https://fr.wikipedia.org/wiki/Gestion_de_versions#Gestion_de_versions_d.C3.A9centralis.C3.A9e). C'est un [logiciel libre](https://fr.wikipedia.org/wiki/Logiciel_libre) et gratuit, créé en 2005 par [Linus Torvalds](https://fr.wikipedia.org/wiki/Linus_Torvalds), auteur du [noyau Linux](https://fr.wikipedia.org/wiki/Noyau_Linux), et distribué selon les termes de la [licence publique générale GNU](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU) version 2. Le principal contributeur actuel de Git, et ce depuis plus de 16 ans, est [Junio C Hamano](https://fr.wikipedia.org/w/index.php?title=Junio_C_Hamano&action=edit&redlink=1).

Depuis les années 2010, il s’agit du [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) le plus populaire dans le développement logiciel et web, qui est utilisé par des dizaines de millions de personnes, sur tous les environnements (Windows, Mac, Linux)[3](https://fr.wikipedia.org/wiki/Git#cite_note-3). Git est aussi le système à la base du célèbre site web [GitHub](https://fr.wikipedia.org/wiki/GitHub), le plus important hébergeur de code informatique.

Git a été spécialement développé et optimisé pour le noyau [Linux](https://fr.wikipedia.org/wiki/Linux). [Linus Torvalds](https://fr.wikipedia.org/wiki/Linus_Torvalds) a commencé son écriture en [avril 2005](https://fr.wikipedia.org/wiki/Avril_2005), pour remplacer le programme propriétaire [BitKeeper](https://fr.wikipedia.org/wiki/BitKeeper). En effet, [Larry McVoy](https://fr.wikipedia.org/wiki/Larry_McVoy) a révoqué la possibilité d'utiliser gratuitement son logiciel, après qu'[Andrew Tridgell](https://fr.wikipedia.org/wiki/Andrew_Tridgell) eut tenté une [rétro-ingénierie](https://fr.wikipedia.org/wiki/R%C3%A9tro-ing%C3%A9nierie) sur les protocoles utilisés par BitKeeper. Les raisons de cette révocation restent toutefois controversées.

La première version a été publiée le 7 avril 2005. Depuis, le développement de cet outil s'est poursuivi. Il a été progressivement amélioré, se voyant doté d'interfaces graphiques, d'interfaces web ou de scripts évolués.

Le Kernel Summit de juillet 2005 a été l'occasion pour Linus Torvalds d'annoncer que les versions suivantes du noyau Linux (à partir de la version 2.6.12) seraient développées à l'aide de Git.

Fin juillet 2005, Linus Torvalds a décidé de confier la maintenance et les évolutions de Git à Junio Hamano, expliquant qu'il avait « toujours dit qu'il ne voulait pas vraiment le maintenir sur le long terme ».

Source : [Git — Wikipédia (wikipedia.org)](https://fr.wikipedia.org/wiki/Git)

## Git Workflow

Il existe plusieurs flux de travail Git, en réalité chaque équipe est libre de créer le sien propre car il s’agit simplement de sa méthode de travail avec Git. Il est important que cette méthode soit adaptée au projet, à la taille de l’équipe, à la culture d’entreprise...

Le workflow centralisé est le plus populaire, les développeurs font des requêtes push (ils envoient leur code vers le dépôt centralisé) et pull (ils récupèrent le code mis à jour par d’autres depuis le dépôt centralisé).

On peut citer à titre d’exemple d’autres types de workflow :

* Création de branches de fonctionnalités
* Gitflow
* Workflow de duplication (fork)

Pour plus d’informations : [workflow git | Atlassian Git Tutorial](https://www.atlassian.com/fr/git/tutorials/comparing-workflows)

Le flux GitHub est un workflow léger basé sur des branches.

En voici les étapes :

* Créer une branche
* Apporter des modifications
* Créer une demande de tirage (Pull Request)
* Traiter les commentaires de Fusirévision
* Fusionner votre demande de tirage
* Supprimer votre branche

Pour plus de détails voir : [GitHub flow - Documentation GitHub](https://docs.github.com/fr/get-started/quickstart/github-flow)

# Pratique

### Pré-requis

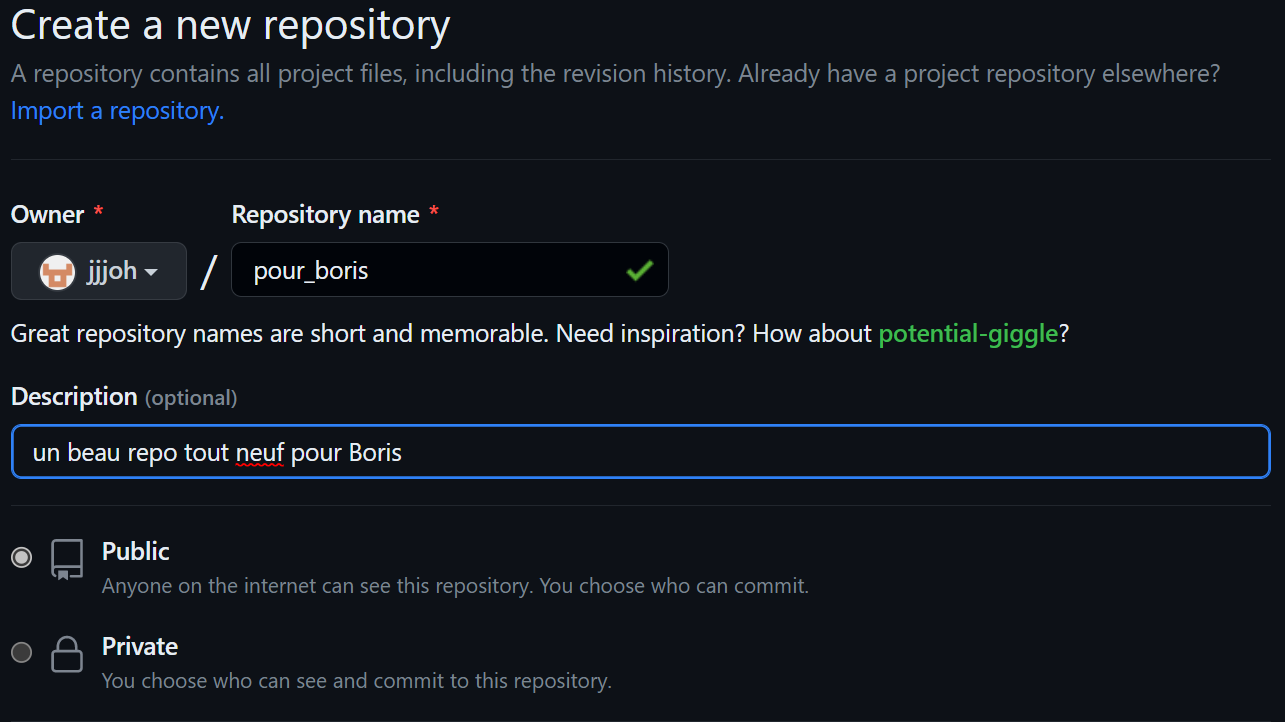
créer /se connecter à son compte GitHub

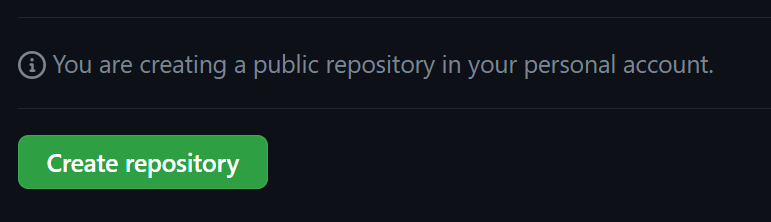
### Etape1

Créer un repository appelé et le rendre public sur le site de github

Méthode 1 avec l’interface graphique depuis l’url :

[Create a New Repository (github.com)](https://github.com/new)





### Etape2

à l'aide du terminal déplacez-vous dans le dossier du projet

### Etape3

initialisé le projet avec git

### Etape4

:ajouté, le projet a git

### Etape5

créer un commit que vous appellerai "version 1"

### Etape6

créer votre branche principale appelée "main"

### Etape7

pushé votre projet

### Etape8

vérifié, que votre projet est bien sur github