

Utilisation de Git

1. Créer un dépôt Git

- Configurer votre Git avec cette commande :

```
git config --global user.name "Votre nom"
```

```
git config --global user.email "votre.email@example.com"
```

Pour créer un nouveau dépôt Git pour le projet sur Gitea il vous suffit de cliquer sur le + en haut à droite de l'application Gitea



Puis de sélectionner la section "Nouveau dépôt". Par la suite cette page s'affichera

Nouveau dépôt

Un dépôt contient tous les fichiers d'un projet, ainsi que l'historique de leurs modifications. Vous avez déjà ça ailleurs ? [Migrez-le ici.](#)

Propriétaire *  barillote
Certaines organisations peuvent ne pas apparaître dans la liste déroulante en raison d'une limite maximale du nombre de dépôts.

Nom du dépôt * Mise en pratique de Git
Idéalement, le nom d'un dépôt devrait être court, mémorisable et unique.

Visibilité Rendre le dépôt privé
Seuls le propriétaire ou les membres de l'organisation, s'ils ont des droits, seront en mesure de le voir.

Description

Modèle

Jeu de labels pour les tickets

.gitignore
De nombreux outils et compilateurs génèrent des fichiers résiduels qui n'ont pas besoin d'être supervisés par git. Composez un .gitignore à l'aide de cette liste des langages de programmation courants.

Licence
Une licence réglemente ce que les autres peuvent ou ne peuvent pas faire avec votre code. Vous ne savez pas laquelle est la bonne pour votre projet ? Comment [choisir une licence.](#)

LISEZMOI
Le README est l'endroit idéal pour décrire votre projet et accueillir des

- Pour initialiser votre projet Git, vous devez ouvrir un terminal et créer un dossier, se place dans ce dossier

```
git init
```

Suite à cela vous avez cette page qui vous sera affichée.

Introduction rapide

Cloner ce dépôt Besoin d'aide pour dupliquer ? Visitez l'aide.

Nouveau fichier Téléverser un fichier **HTTPS** SSH https://gitea.btssio-poitiers.fr/barillote/Premier_Depot.git

Création d'un nouveau dépôt en ligne de commande

```
touch README.md
git init
git checkout -b main
git add README.md
git commit -m "first commit"
git remote add origin https://gitea.btssio-poitiers.fr/barillote/Premier_Depot.git
git push -u origin main
```

Soumission d'un dépôt existant par ligne de commande

```
git remote add origin https://gitea.btssio-poitiers.fr/barillote/Premier_Depot.git
git push -u origin main
```

- Pour cloner le dépôt existant

```
git clone https://gitea.btssio-poitiers.fr/barillote/Premier_Depot
```
- Pour faire un git flow

```
git flow init
```

```
barillote@A103-02-LINUX:~/Documents/Premier_Depot$ git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [/home/barillote@stsio.lan/Documents/Premier_Depot/.git/hooks]
```

- Pour aller dans une branche

```
git checkout -b main
```
- Pour créer le fichier index.html

```
touch index.html
```
- Pour ouvrir le fichier index.html

```
nano index.html
```

- Pour ajouter le fichier index.html à la zone de staging
`git add index.html`
- Pour valider la première commit
`git commit -m "index.html create"`
- Pour envoyer les commits
`git push`
- Pour sélectionner tous les fichiers qui ont changés dans la liste d'attente
`git add .`
- Pour créer le fichier pour exclure la sauvegarde de certains fichiers (fichiers temporaires, fichiers contenant des mots de passe,...)
`touch ./gitignore`
- Pour configurer le .gitignore
`nano ./gitignore`

Configurez le comme ceci :

```
GNU nano 7.2                               ./gitignore *
*.log
*.tmp
```

Cela vous permettra d'ignorer l'enregistrement de fichiers temporaires ainsi que les fichiers de log.

- Pour créer une branche s'appelant header
`git branch header`
- Pour se placer dans la branche header
`git checkout header`
- Pour fusionner la branche header dans la branche main
`git checkout main`
`git merge header`
- Pour envoyer le commit
`git commit -m "header"`
- Pour push le commit
`git push`