

TP1

Démarrez votre projet avec GitHub

- 1) GitHub est un service en ligne qui va vous permettre d'héberger vos dépôts distants.
Pour créer votre compte GitHub, rendez-vous sur la page <https://github.com/> et cliquez sur Sign up (en haut à droite de l'écran).
- 2) On vous demandera alors de renseigner un e-mail, un mot de passe et un nom d'utilisateur.
- 3) Un code de vérification vous sera envoyé sur votre adresse e-mail afin de confirmer votre identité.
Et voilà, vous êtes à présent inscrit sur GitHub ! Par défaut, GitHub est gratuit. Mais sachez qu'il existe également des offres payantes si vous décidez de passer à la vitesse supérieure.

GitHub est assez facile à prendre en main et simple d'utilisation.

Votre tableau de bord

Vous pouvez consulter votre **tableau de bord** personnel pour :

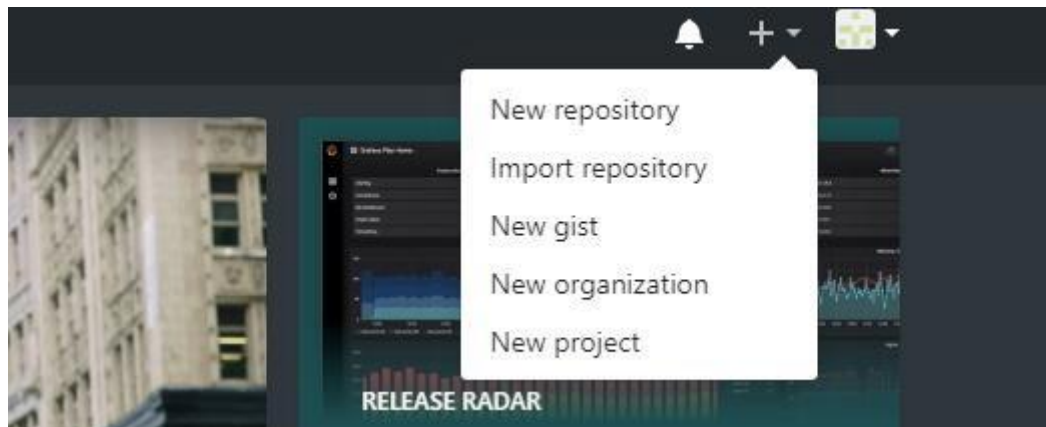
- suivre les problèmes et extraire les demandes sur lesquelles vous travaillez ou que vous suivez ;
- accéder à vos principaux repositories et pages d'équipe ;
- rester à jour sur les activités récentes des organisations et des repositories auxquels vous êtes abonné.

L'interface repository

L'interface Repositories est l'emplacement où vous pourrez créer et retrouver vos dépôts existants.

Pour créer un projet, il suffit de cliquer sur "Start a project".

- 4) Pour mettre votre projet sur GitHub, vous devez créer un **repository** (ou dépôt en français) dans lequel il pourra être installé.
- 5) Cliquez sur le "+" dans le coin supérieur droit, pour faire apparaître l'option New repository.



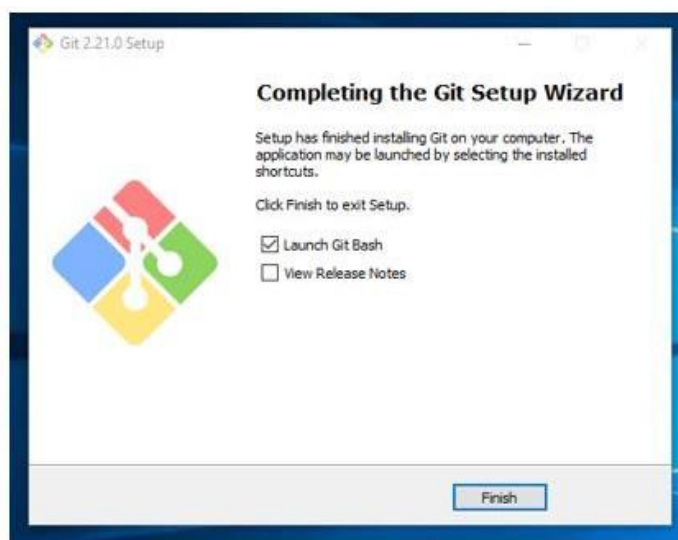
- 6) Choisissez un nom simple pour votre dépôt. Dans ce TP, nous utiliserons "Project1". Puis, choisissez si vous souhaitez créer un dépôt public ou privé. Choisir l'option « public ».

Installez Git sur votre ordinateur

- 1) Choisissez et téléchargez la version de Git qui correspond à votre installation : MacOS, Windows ou Linux/Unix.
<https://git-scm.com/downloads>
- 2) Exécutez le fichier que vous venez de télécharger.
- 3) Appuyez sur Suivant à chaque fenêtre puis sur **Installer**. Lors de l'installation, laissez toutes les options par défaut, elles conviennent bien.

Si vous êtes sous Windows : cochez ensuite Launch Git Bash. Pour les utilisateurs de Mac ou Linux, votre terminal suffira amplement.

Git Bash est l'interface permettant d'utiliser Git en ligne de commande.



- 4) Git Bash se lance.

```
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

$
```

Pour travailler sur Git, vous devez créer un dépôt local, c'est-à-dire un dossier dans lequel toutes vos modifications seront enregistrées. C'est ce qu'on appelle **initialiser un dépôt Git**.

- 5) Configurez votre identité : La première chose à faire est de configurer votre identité. Pour cela, vous allez entrer dans le monde des lignes de commande !

Commencez par renseigner votre nom et votre adresse e-mail. **C'est une information importante** car vous en aurez besoin pour toutes vos validations dans Git :

```
git config --global user.name "John Doe"
git config --global user.email "johndoe@example.com"
```

Grâce à l'option `--global`, vous n'aurez besoin de le faire qu'une fois.

Si vous souhaitez, pour **un projet spécifique**, changer votre nom d'utilisateur, vous devrez repasser cette ligne mais sans le `--global`.

Pour vérifier que vos paramètres ont bien été pris en compte, et vérifier les autres paramètres, il suffit de passer la commande `git config --list` :

```
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

$ git config --list
core.symlinks=false
core.autocrlf=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
pack.packsizelimit=2g
help.format=html
http.sslcainfo=/bin/curl-ca-bundle.crt
sendemail.smtpserver=/bin/msmtp.exe
diff.astextplain.textconv=astextplain
rebase.autosquash=true
```

- 6) Configurez les couleurs : Il est recommandé d'activer les couleurs afin d'améliorer la lisibilité des différentes branches. Pour cela, passez ces trois lignes dans Git Bash :

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

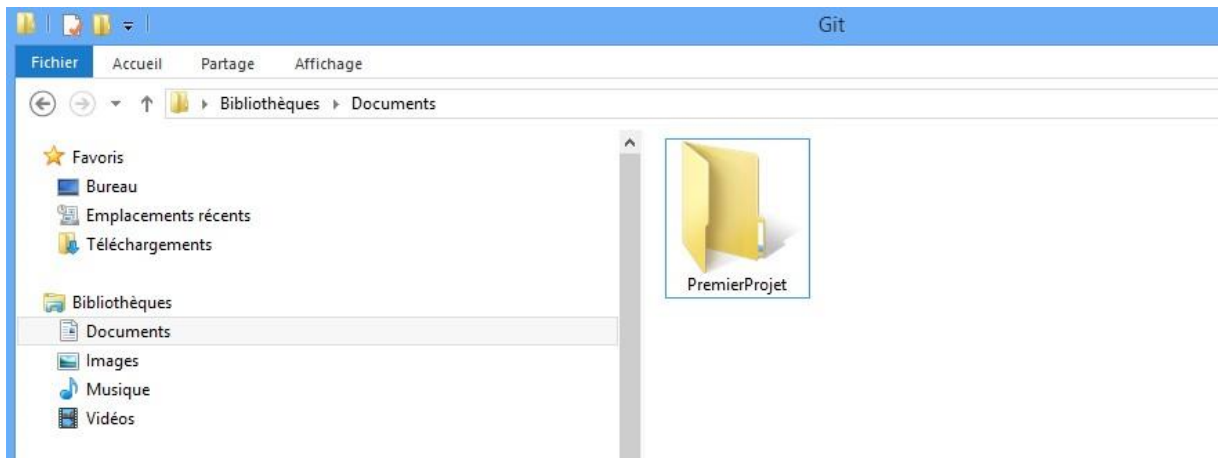
- 7) Configurez l'éditeur : Par défaut, Git utilise Vim comme éditeur et Vimdiff comme outil de merge. Vous pouvez les modifier en utilisant :

```
git config --global core.editor "notepad++"
git config --global merge.tool vimdiff
```

- 8) Créez votre dépôt local : Maintenant que vous avez configuré les paramètres de base, vous pouvez **créer votre dépôt local** :

Dans un premier temps, créez un dossier sur votre ordinateur :

- Allez dans "Document".
- Créez le dossier "PremierProjet".
- Accédez à votre dossier.



- 9) Ensuite, lancez deux lignes de commande dans Git Bash :

```
johndoe ~
$ cd Documents/PremierProjet
johndoe ~/Documents/PremierProjet
$ git init
Initialized empty Git repository in c:/users/JohnDoe/Documents/PremierProjet/
```

La première ligne permet de vous positionner dans le dossier que vous venez de créer sur l'ordinateur. La seconde ligne va initialiser ce "simple dossier" comme un dépôt.