

GitHub

Pour la mise en œuvre du projet, nous utilisons GitHub, et voici quelques astuces afin de l'utiliser correctement et pouvoir travailler avec !

Windows :

Vous êtes sûrement entrain de travailler sur un windows pour la création de votre site vitrine ou bien pour le développement du site, honnêtement peu m'importe mais l'idéal serait sur du linux, et github est un bon moyen de remplacer le ssh pour travailler aussi sur votre machine linux mais comment fait-on alors ?

GitHub nous propose un moyen d'actualiser un répertoire présent sur github (celui que nous avons créé par exemple) directement sur votre terminal de commande. Vous êtes maintenant à l'aise avec les terminaux et à l'exécution de commande dans celui-ci mais avant de commencer étant donné que nous sommes sur un windows nous avons quelques installations à faire avant.

Premièrement rendez-vous sur ce site : <https://git-scm.com/downloads>

installer la version standalone pour windows en 64 bits, lors de l'installation faites bien attention à cocher la case "**Git Bash**", une fois l'installation faites normalement vous aurez un git hub sur votre pc et le plus important les commandes git sur votre cmd...

Maintenant que vous avez vos commande Git sur votre bash vous allez maintenant décider vous même d'un emplacement sur votre ordinateur pour y mettre votre Git, alors je vous déconseille de le mettre sur le bureau en situation d'entreprise et en général lorsque nous travaillons sur un projet en informatique on essaye de le mettre dans le disque "C:\\" directement, bien que le bureau sois dans le disque "C:", donc je vous invite à ouvrir votre **explorateur de fichier**, de vous rendre dans **ce PC** et

cliquer sur votre disque et d'y créer un dossier "Git Hub" (ou le nom que vous voulez)

Maintenant que votre dossier est créé, ouvrez un inviteur de commande, déplacer vous dans votre dossier (**cd "C:\Git Hub\"** ou **cd C:\Git Hub**)

Une fois dedans, entrez cette commande :

```
git clone https://github.com/rba95/ITS
```

vous pouvez ensuite vérifier directement avec votre explorateur de fichier si il est bien là ou faire un **"ls"** dans votre terminal.

ensuite rendez-vous sur Git Hub > paramètre > developper settings > Personnel access token > Generate a classic token > et cocher toute les cases en y mettant une note (j'ai mis projet personnellement)

Une clé sera générée, copier la dans un fichier texte (temporairement)

Maintenant que vous avez votre clé token rendez-vous sur votre terminal encore une fois et entrez ces commandes la :

```
git config --global user.name "Votre nom git hub"  
git config --global user.email "Votre mail associer"  
git config --global credential.helper store
```

ces commandes vous permettront à l'avenir à ne plus entrer vos nom git hub et mot de passe peu importe le projet.

ensuite vérifier bien que sur votre terminal vous êtes sur le répertoire git hub que nous avons cloner et entrez ces commandes :

```
git status
```

permet de voir si vous êtes bien à jour (à faire a chaque utilisation pour être sur)

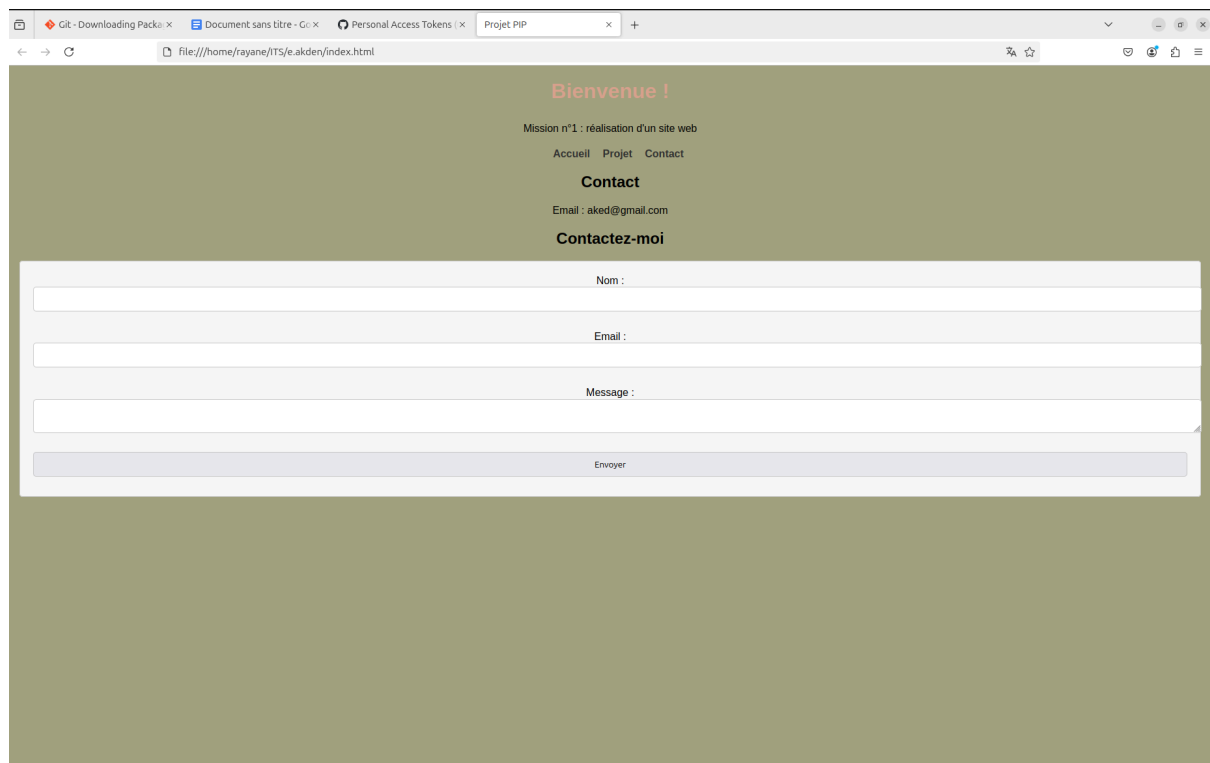
```
git pull
```

permet de récupérer les mises à jour apporter au dossier, par exemple ici nous

avons Eda qui a apporté du changement :

```
rayane@thinkpad:~/ITS$ git pull
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Dépaquetage des objets: 100% (8/8), 3.07 Kio | 3.07 Mio/s, fait.
Depuis https://github.com/rba95/ITS
 8857a6e..c5db2ea  main    -> origin/main
Mise à jour 8857a6e..c5db2ea
Fast-forward
 README.md          | 2 +-
 e.akden/index.html | 39 ++++++
 e.akden/style.css  | 93 ++++++
 3 files changed, 133 insertions(+), 1 deletion(-)
 create mode 100644 e.akden/index.html
 create mode 100644 e.akden/style.css
rayane@thinkpad:~/ITS$
```

et j'ai donc accès à son site :



ensuite nous avons :

```
git add -A
```

permet d'ajouter les fichiers à envoyer peu importe où vous vous situez dans le répertoire, par contre si vous voulez ajouter un fichier en particulier vous avez juste à mettre le nom du fichier exemple :

```
git add Fichier.txt
```

ensuite nous entrons cette commande :

```
git commit -m "Ajout d'un fichier texte pour explication"
```

Vous l'aurez compris cela permet d'expliquer les modifications ajouter.

```
git push
```

permet d'envoyer les modifications local directement sur le github, normalement lors de la première utilisation de la commande le nom de votre username sera demander (ce sera la seule et unique fois comme pour le mot de passe) :

```
rayane@thinkpad:~/ITS$ git add .
rayane@thinkpad:~/ITS$ git push
Username for 'https://github.com': rba95
Password for 'https://rba95@github.com':
Everything up-to-date
rayane@thinkpad:~/ITS$
```

dans le mot de passe vous mettrez le token générer que vous avez stocker (je vous conseille de le garder quand même en cas d'erreur de ma part sinon vous devrez générer un nouveau token pour effectuer des actions)

!! attention sur le screen je n'ai pas fais le git commit -m "...” je n'ai donc pas retrouvé mon fichier ajouter en local, alors faites le bien a chaque ajout de fichier et expliquer votre changement !!

et voici un exemple de comment supprimer un fichier dans un repertoire :

```
rayane@thinkpad:~/ITS/r.barkallah$ rm test.txt
rm : supprimer 'test.txt' qui est protégé en écriture et est du type « fichier » ? oui
rayane@thinkpad:~/ITS/r.barkallah$ ls
README
rayane@thinkpad:~/ITS/r.barkallah$ git add .
rayane@thinkpad:~/ITS/r.barkallah$ git commit -m "Suppression test"
[main 7a473c0] Suppression test
1 file changed, 1 deletion(-)
delete mode 100644 r.barkallah/test.txt
rayane@thinkpad:~/ITS/r.barkallah$ git push
Énumération des objets: 5, fait.
Décompte des objets: 100% (5/5), fait.
Compression par delta en utilisant jusqu'à 14 fils d'exécution
Compression des objets: 100% (2/2), fait.
Écriture des objets: 100% (3/3), 295 octets | 295.00 Kio/s, fait.
Total 3 (delta 1), réutilisés 0 (delta 0), réutilisés du pack 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/rba95/ITS
bd87d12..7a473c0 main -> main
```

```
rm test.txt  
git add .  
git commit -m "Suppresion"  
git push
```

Voici tout ce que vous avez besoin de savoir pour travailler dans de bonne condition avec git hub, en cas de question n'hésitez pas à me les poser ou de vous renseignez sur internet !

Pour la prochaine étape vous allez héberger votre site web sur un serveur web apache2 ;)

cette fois-ci je veux que vous le fassiez en linux obligatoirement !

petit tips :

Configurez votre vm linux en accès par pont pour que vous puissiez y accéder via l'ip de votre vm !