

LIGNIERE

Timothée

BTS2 SIO

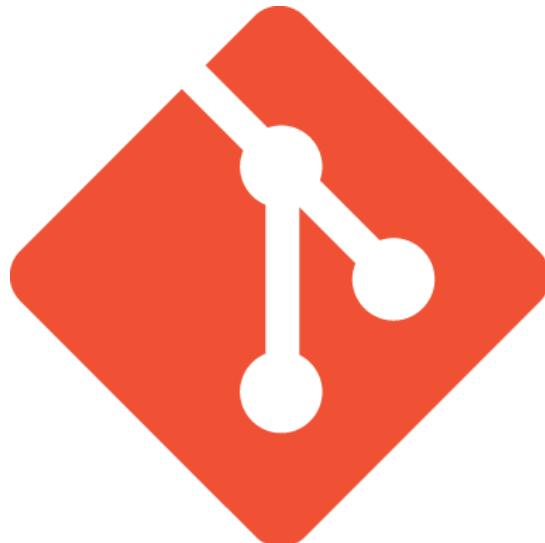
Date de création

06/12/2023

Date de maj

07/12/2023

Git - Github



Sommaire

1 - Problématique.....	2
2 – Prérequis.....	3
2.1 – Installation de git.....	3
2.2 – Token d'authentification Github.....	3
3 – Utilisation de git.....	6
3.1 – Téléchargement d'un dépôt.....	6
3.2 – Exporter vers un dépôt pour la première fois.....	6
3.3 – Apporter des modifications à un dépôts.....	7
4 - Annexes.....	8

1 - Problématique

Pour un très grand nombre de projets, il est nécessaire d'interagir avec des sites tels que Github ou Gitlab car les logiciels se trouvent sur ces plateformes, aussi cela peut aussi être une plateforme permettant d'accéder facilement à de multiples fichiers de configurations.

Enfin, ces deux plateformes laissent leurs utilisateurs déposer leurs propres fichiers dans la limite du raisonnable pour aider d'autres utilisateurs ou y accéder depuis n'importe quelles machines reliées à Internet.

Nous allons donc nous intéresser à la commande qui permet d'interagir avec ces deux plateformes, git. Je tiens à préciser que je me concentrerai sur l'utilisation de git avec Github.

2 – Prérequis

À partir de ce stade, je considère que vous avez créé un compte sur Github de la manière que vous préférez pour les étapes suivantes.

2.1 – Installation de git

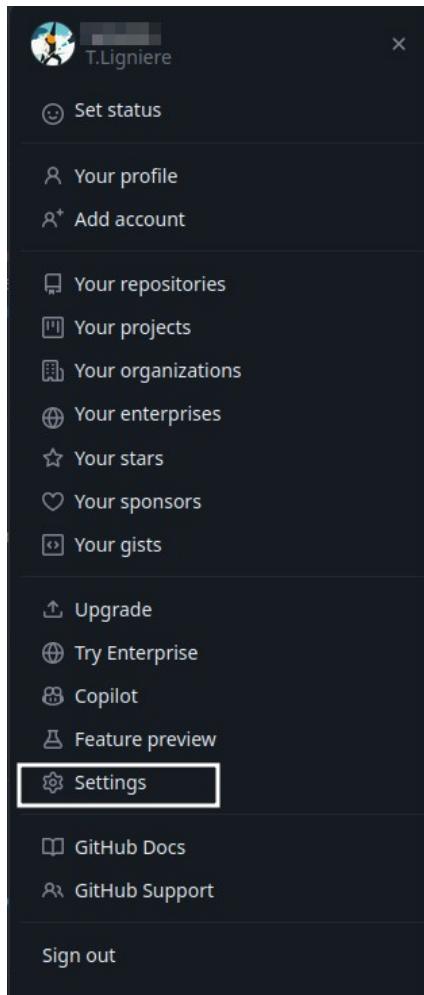
L'installation de git en lui-même est la partie facile :

```
sudo apt install -y git-all
```

2.2 – Token d'authentification Github

Depuis plusieurs années, Github ne permet plus d'authentification par mot de passe lors de l'utilisation de git pour exporter des informations ou télécharger depuis un dépôt privé, il est donc nécessaire de créer un token pour les remplacer.

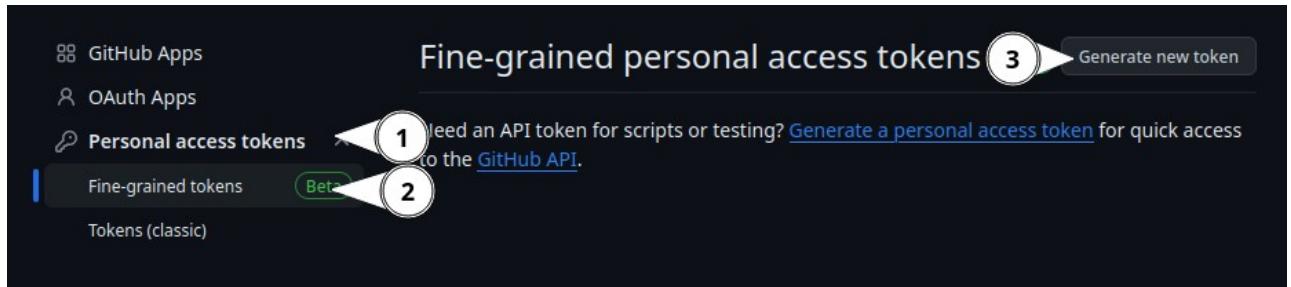
Pour cela, connectez-vous à Github (<https://github.com/>), cliquer sur votre image de profil puis aller dans « Settings » :



Une fois dans les paramètres de compte, dirigez-vous vers l'onglet « Developper settings », il se trouve tout en bas des différents paramètres :

Developer settings

Puis choisissez « Personal Access Tokens » > « Fine-grained tokens » et enfin « Generate new tokens » :



Donnez ensuite un nom à votre token et choisissez le temps d'expiration du token, penser à modifier « Repository access », si vous avez des dépôts privés, votre token ne vous permettra pas d'y accéder sans modifier ce paramètre :

New fine-grained personal access token Beta

Create a fine-grained, repository-scoped token suitable for personal API use and for using Git over HTTPS.

Token name *
Full access token ✓

A unique name for this token. May be visible to resource owners or users with possession of the token.

Expiration *
30 days ▼ The token will expire on Fri, Jan 5 2024

Description

What is this token for?

Resource owner
 Splatda ▼

Repository access

Public Repositories (read-only)

All repositories
This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

Only select repositories
Select at least one repository. Max 50 repositories.
Also includes public repositories (read-only).

Vous pouvez rajouter des permissions si vous le jugez nécessaire dans la catégorie « Permissions ».

Si lors d'une utilisation de git, vous obtenez une erreur 403 par exemple un git push alors cela signifie que votre token n'a pas suffisamment de permissions, vous pouvez cependant tout à fait les modifier en allant dans les paramètres du token.

Une fois que vous aurez complété la configuration du token, pensez bien à conserver le token que vous venez de générer dans un fichier et d'aussi noter sa date d'expiration pour savoir quand vous ne pourrez plus l'utiliser.

Si vous perdez le token ou s'il devient expiré, répéter la procédure.

3 – Utilisation de git

3.1 – Téléchargement d'un dépôt

Nous allons commencer une des commandes les plus utilisées permettant de télécharger un dépôt complet **git clone**.

```
git clone <URL du dépôt>
```

3.2 – Exporter vers un dépôt pour la première fois

Là où télécharger depuis Github est simple, exporter est plus complexe.

Pour commencer, créer un nouveau dépôt depuis le site, puis retourner sur le terminal de commande de votre machine contenant les informations que vous voulez exporter.

```
mkdir <Nom du dossier que vous créez>
cd <Nom du dossier créé à l'instant>
```

Désormais, nous allons indiquer à git que nous créons un nouveau dépôt :

```
git init
```

Configurer vos identifiants pour les indiquer à git :

```
git config --global user.email <Votre adresse mail sur le compte github>
git config --global user.name <Nom d'utilisateur sur github>
```

Puis indiquer les fichiers que vous voulez rajouter au dépôts :

```
git add <Nom/Chemin vers le fichier/dossier>
```

Si vous avez tout positionné dans un même dossier auparavant, vous pouvez utiliser l'argument « . » qui indique que vous voulez rajouter tout ce qui se trouve dans ce dossier.

Vous ne pouvez pas exporter des fichiers plus gros que 50Mo avec un compte gratuit.

Une fois cela fait, nous allons indiquer à git que vous avez complété vos ajouts en les plaçant sous un nom :

```
git commit -m <Nom de votre action>
```

Maintenant, nous allons indiquer à git vers quels dépôts envoyer les informations que vous lui avez données

```
git remote add <Nom de dépôt> <Lien du dépôt>
```

Enfin, envoyer les informations au dépôt :

```
git push <Nom de dépôt>
```

Github vous demandera votre nom d'utilisateur et un mot de passe, vous pouvez mettre votre nom d'utilisateur sans aucun problème, **cependant, pour le mot de passe, vous devez utiliser le token que vous avez généré au préalable sur le site.**

3.3 – Apporter des modifications à un dépôt

Apporter des modifications à un dépôt peut être destructeur si l'on ne fait pas attention à ce que l'on fait, pour commencer, retourner dans le dossier que vous aviez créés ou réutiliser la commande suivante dans un dossier vide :

```
git init
```

Puis importer le dépôt depuis github **!\\ Si vous avez des fichiers ou dossiers déjà existant dans le dossier où vous vous trouvez, il est fort probable qu'ils soient supprimés (mis à part si ils sont eux mêmes présents dans le dépôt /\\ :**

```
git pull
```

Si vous avez fait git init, vous aurez sûrement besoin de refaire la configuration du dépôt distant et de vos informations d'utilisateur

Puis apporter les modifications que vous vouliez faire à votre dépôt dans le fichier dans lequel vous vous trouvez.

Une fois que vous avez fait ce que vous jugez nécessaire, faites un commit :

```
git commit -m « <Nom de commit> »
```

Et enfin envoyer les modifications vers le dépôt :

```
git push <Lien du dépôt>
```

Vous aurez besoin de votre identifiant github et de votre Token.

Répéter la procédure à chaque fois où vous le jugez nécessaire.

Dans certains cas, les pushes doivent être confirmés par le propriétaire du dépôt github.

4 - Annexes

Fiche de recette

Vérification de l'opérationnalité de la solution mise en œuvre : *Git*

Description du test :

1. Télécharger un dépôt
2. Exporter des fichiers/dossiers
3. Mettre à jour un dépôt

Résultats Attendus :

1. Dépôt téléchargé
2. Fichiers et dossiers exporter sur *github* bien accessible
3. Mise à jour du dépôt réussi

Réception Globale : Git	Date:07/12/2023
Auteurs: Timothée LIGNIERE	
Reçu :	<input type="checkbox"/>
Reçu avec réserve :	<input type="checkbox"/>
Refusé :	<input type="checkbox"/>
Commentaire :	

Recette étape par étape *

* (pour chaque étape, vous devez élaborer dans un fichier distinct un scénario détaillé à faire appliquer au « client » venant valider votre solution)

Réception Etape 1: Télécharger un dépôt à l'aide de git clone pour vérifier que la commande est bien fonctionnel sur la machine	
Reçu :	<input type="checkbox"/>
Reçu avec réserve :	<input type="checkbox"/>
Refusé :	<input type="checkbox"/>
Commentaire :	

Réception Etape 2 : Exporter des dossiers et fichiers vers son dépôt personnel sur github pour vérifier le bon fonctionnement du token généré sur le site et l'exportation d'information	
Reçu :	<input type="checkbox"/>
Reçu avec réserve :	<input type="checkbox"/>
Refusé :	<input type="checkbox"/>
Commentaire :	

Réception Etape 3 : Mettre à jour un dépôt pour sécuriser ses informations ou aider d'autres dépôts d'autres projets	
Reçu :	<input type="checkbox"/>
Reçu avec réserve :	<input type="checkbox"/>
Refusé :	<input type="checkbox"/>
Commentaire :	

