□□□□□ Publier et collaborer

Maintenant on va voir comment publier son code sur une "git forge" tel que GitHub, Gitlab, Gitea, Codeberg, ou autre.

Tout d'abord il faut se créer un compte sur l'un de ces sites. Dans ce tutoriel on va utiliser Gitlab, car on va le réutiliser dans le chapitre suivant sur l'automatisation.

Ajouter une clé SSH pour ne pas devoir entrer son mot de passe dans Git

On pourrait simplement donner l'URL de notre projet à Git et lui dire de publier le code. Le problème c'est qu'a chaque fois que l'on voudra faire ça, Git va redemmander notre nom d'utilisateur et mot de passe.

Une méthode beaucoup plus sécurisée et beaucoup plus simple est d'utiliser des *clés SSH*.

Si ce n'est pas encore le cas, vous pouvez en créer en utilisant la commande suivante :

```
# Remplissez les champs demandés ou faites ENTER pour avoir les paramètres par défault
# Faites attention à ne pas mettre de mot de passe en appuyant sur ENTER 2 fois.
ssh-keygen
```

Ensuite vous pouvez afficher votre "clé publique" (ce n'est pas confidentiel et c'est ce que Gitlab va utiliesr pour vérifier votre identité)

```
# La commande "cat" lit un fichier. Ce fichier est notre clé publique
cat ~/.ssh/id_rsa.pub
```

Enfin on peut aller dans les paramètres de gitlab et ajouter cette clé.

Page d'ajout de la clé SSH sur Gitlab

Image not found or type unknown

Publier un projet existant sur Git

Lien pour la création d'un repo sur Gitlab

Image not found or type unknown

Page de création d'un repo sur Gitlab

Image not found or type unknown

On peut choisir si on veut que le repo soit privé, ou public. Il est aussi préférable de ne pas activer le "README" automatique

Une fois que le "repo" est créé sur le site on peut soit simplement utiliser les commandes données, ou juste copier le lien SSH. Dans ce cas on va faire cela:

Copier le lien SSH

Image not found or type unknown

```
# On suppose que vous êtes déjà dans le projet, mais sinon ouvrez le terminal ou cd comme vu dans la section précédente
git remote add origin git@gitlab.com:julien.lejoly2712/mon-super-projet.git
```

Une fois le lien du "remote" ajouté. On peut maintenant "push" (publier) le code sur le site:

```
git push origin master
```

Et maintenant, sans avoir mis aucun mot de passe, mais en restant sécurisé, le code devrait être accessible sur la page du projet !

La page du projet maintenant

Image not found or type unknown

Ajouter un collaborateur au projet

Maintenant si on veut ajouter un autre collaborateur sur le projet, on peut aller dans les informations du projet puis dans "Membres"

Screenshot du menu

Image not found or type unknown

Ensuite on peut cliquer sur "Inviter des membres" et inviter quelqu'un sous le rôle de "développeur".

Screenshot du formulaire

Image not found or type unknown

Maintenant lea collaborateur•ice peut télécharger le projet en utilisant la commande `git clone`

```
# L'addresse est l'addresse SSH prise plus tot.
# Si on veut juste télécharger le code sans la possibilité de le modifier, on peut utiliser le lien HTTPS
git clone git@gitlab.com:julien.lejoly2712/course-book.git
cd course-book
```

A présent on peut utiliser Git comme vu précédemment. Mais la différence c'est qu'il faut de temps en temps aussi re-synchroniser le projet en utilisant `git pull`

```
git pull
```

Collaborer sur un projet avec des "forks"

Si vous n'êtes pas un membre d'un projet mais souhaitez quand même contribuer, vous pouvez faire un "fork".

Un fork est une copie d'un projet. Vous pouvez ainsi y faire des modifications puis proposer d'ajouter (merge) les modifications dans le projet de base.

Dans cet exemple on va faire un fork du projet Gitlab. Pour cela il suffit de cliquer sur le bouton "fork" en haut à droite.

Screenshot d'un fork de gitlab

Ensuite on peut copier le lien SSH vers notre fork :

Copier le lien vers le fork

Enfin, on peut télécharger le code et y faire nos changements.

```
git clone git@gitlab.com:julien.lejoly2712/gitlab.git
cd gitlab

# Pour l'exemple on va ajouter une ligne à la fin du fichier README.md
echo "Ceci est une ligne inutile" >> README.md

# On va ensuite ajouter les changements (le -a dans la commande commit va "add" tous les fichiers suivi qui ont été modifiés)
git commit -am "J'ai ajouté une ligne inutile"
```

```
# On peut ensuite publier les changements sur notre fork
git push origin master
```

Une fois cela fait, on peut maintenant proposer d'ajouter nos changements sur le projet d'origine en créant un "merge request"

> **ATTENTION**: Tout ceci est à titre d'**exemple**, le projet Gitlab est un vrai projet et les développeurs ont d'autre choses à faire que gérer des "merge requests" inutiles. **Merci de ne pas réellement faire ceci**

On peut donc créer une nouvelle "merge request" sur Gitlab.

Screenshot création merge request
Image not found or type unknown

Ensuite on précise quel "branche" (on va voir en détails les branches après), et quel projets on va fusionner. Dans ce cas on précise que `gitlab-org/gitlab` est la destination de notre proposition de fusion.

Screenshot merge request
Image not found or type unknown

Enfin on peut ajouter une description de nos changements pour finir notre merge request.

Screenshot du formulaire
Image not found or type unknown

A présent, les membres du projet de base peuvent ajouter tous vos changements en un clic, si tout se passe bien (on va voir les conflits Git plus tard).

Screenshot pov membre
Image not found or type unknown

Revision #1
Created 26 April 2023 19:09:21 by SnowCode
Updated 26 April 2023 19:09:21 by SnowCode