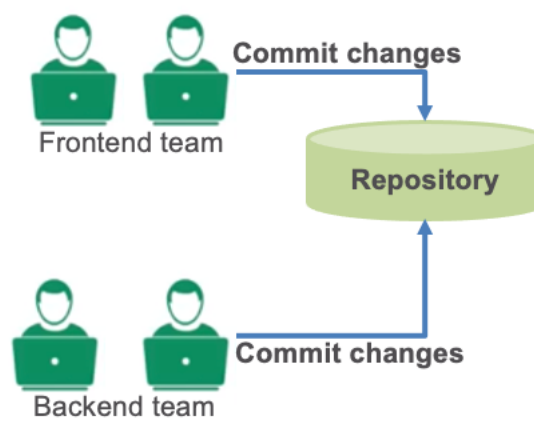


## Git avec un dépôt distant

Précédemment, vous avez vu comment créer un dépôt Git local. Cela vous permet de sauvegarder et de gérer des versions de votre projet. Cependant, cela n'est pas suffisant pour un projet collaboratif où plusieurs développeurs y contribuent.

Donc il faut utiliser ce qu'on appelle un dépôt distant. Ainsi, les codes sources d'un projet sont centralisés sur un dépôt (repository) et les membres qui y travaillent envoient (poussent) leurs modifications sur ce dépôt.



Pour mettre en place un tel dépôt, vous allez utiliser la plateforme Github.

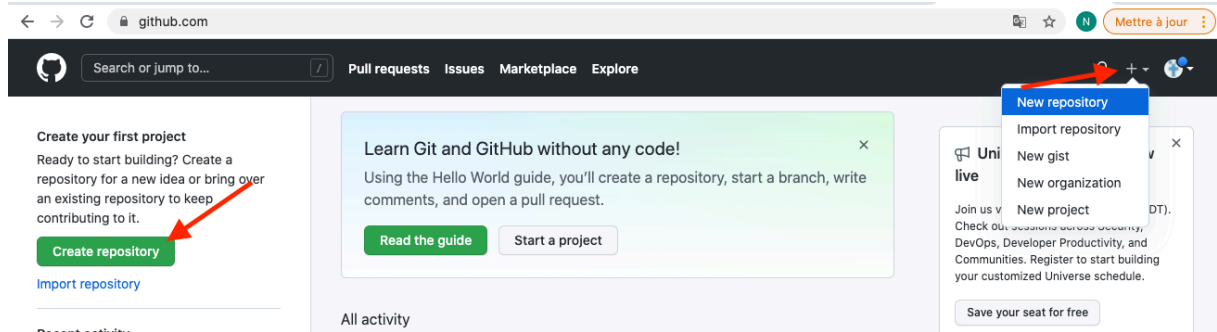
Github est un site Web qui offrent des supports pour le partage de code sources entre plusieurs développeurs. Il est associé à l'outil Git pour la gestion des versions. En quelque sorte, Github complète Git par des capacités de collaborations pour la mise en place d'un projet.

### 1. Préparation d'un dépôt sur Github :

Vous allez compléter ce que vous avez fait lorsque vous avez mis en place un répertoire Git en local par un dépôt Github. Si vous avez déjà un compte GitHub que vous voulez utiliser pour ce TD, vous pouvez passer à l'étape 2.

Donc, si vous n'avez pas de compte sur github ou si vous voulez créer un juste pour faire vos test pour ce TD, allez sur <https://github.com/> et créez un compte (<https://github.com/signup?source=login>).

Depuis votre compte, créez un nouveau « repository » via l'un des deux moyens indiqués par une flèche rouge :



Donnez un nom à votre repository. Ici on l'appelle DemoTest, et c'est un projet public.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner \*

 nggei ▾

Repository name \*

/ DemoGit ✓

Great repository names are short and memorable. Need inspiration? How about [furry-garbanzo](#)?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more](#).

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

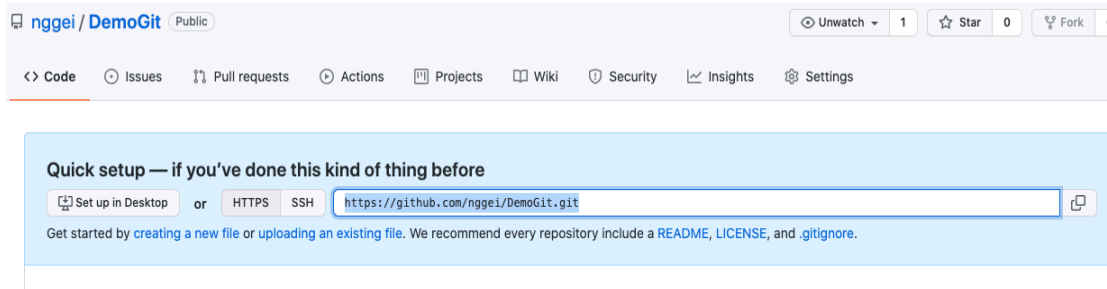
Create repository

## 2. Utilisation du dépôt distant :

Reprenez le répertoire Git local précédemment créé. Vous allez pousser le contenu de votre répertoire dans le répertoire distant. Cela s'appelle un *push*.

Pour faire cela, il faut récupérer l'URL de votre repository Github. Il suffit d'ouvrir votre repository, puis copier l'URL.

TD PDLA  
4IR  
INSA de Toulouse  
[guermouc@insa-toulouse.fr](mailto:guermouc@insa-toulouse.fr)



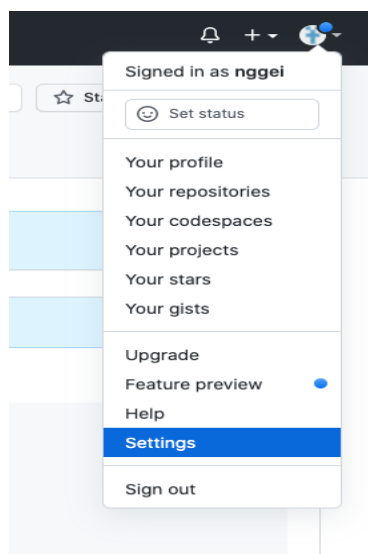
Une fois vous avez récupéré l'URL de votre repository, lancez les commandes suivantes pour pousser tous les commits déjà effectués sur le dépôt :

- **git remote add origin l'url\_copiée** (ici on a lancé la commande : *git remote add origin <https://github.com/nggei/DemoGit.git>*) : associer votre répertoire Git au dépôt distant en créant un alias *origin* (vous pouvez utiliser n'importe quel nom)
- **git push -u origin master** : pousser les modifications de la branche master sur le dépôt distant spécifié avec l'alias origin déjà crée (pour éviter de taper à chaque fois toute l'URL du dépôt distant, on utilise l'alias)

Vous allez certainement obtenir le signalement suivant :

Depuis Aout 2021, l'authentification est basée sur l'utilisation des token : <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token#creating-a-token>

Vous l'avez compris, il faut créer un token et l'utiliser à la place de votre mot de passe utilisateur. Pour faire cela, allez sur votre profil/setting sur Github :



TD PDLA  
4IR  
INSA de Toulouse  
[guermouc@insa-toulouse.fr](mailto:guermouc@insa-toulouse.fr)

Puis allez dans *Developer settings*

The screenshot shows the GitHub profile settings page for a user named 'nggei'. On the left is a sidebar with a list of settings categories: Account settings, Profile, Account, Appearance, Account security, Billing & plans, Security log, Security & analysis, Sponsorship log, Emails, Notifications, Scheduled reminders, SSH and GPG keys, Repositories, Packages, Organizations, Saved replies, Applications, Developer settings (highlighted with a red arrow), Moderation settings, Blocked users, and Interaction limits. The main content area is titled 'Public profile' and contains several input fields for profile information: Name, Public email (with a dropdown to 'Select a verified email to display'), Bio, URL, Twitter username, Company, and Location. To the right of these fields is a 'Profile picture' section with a circular placeholder image and an 'Edit' button. At the bottom of the main content area is a green 'Update profile' button. A red arrow points from the 'Developer settings' link in the sidebar to the 'Developer settings' link in the main content area.

Allez dans *Personal access tokens*

The screenshot shows the GitHub Developer settings page. At the top, there is a breadcrumb trail: 'Settings / Developer settings'. Below this is a sidebar with three links: 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' (highlighted with a red arrow). The main content area is titled 'GitHub Apps' and contains a 'New GitHub App' button. Below the button is a paragraph of text: 'Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).' A red arrow points from the 'Personal access tokens' link in the sidebar to the 'Personal access tokens' link in the main content area.

TD PDLA  
4IR  
INSA de Toulouse  
[guermouc@insa-toulouse.fr](mailto:guermouc@insa-toulouse.fr)  
Cliquez sur *Generate new token*

[Settings](#) / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

### Personal access tokens

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Generate new token

Donnez un nom à votre jeton. Choisissez la durée d'expiration et cocher les autorisations accordées via ce jeton. Ici on a tout coché pour donner tous les droits à ce token.

GitHub Apps

OAuth Apps

Personal access tokens

### New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

TokenDemo

What's this token for?

**Expiration \***

90 days

 The token will expire on Mon, Jan 24 2022

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input checked="" type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input checked="" type="checkbox"/> read:org	Read org and team membership, read org projects

Sauvegardez la valeur du jeton généré. C'est ce qu'il faudrait utiliser à la place d'un mot de passe utilisateur. Comme vous pouvez le lire, vous ne pourrez plus le récupérer ultérieurement après fermeture. Sinon, il faudrait générer un nouveau.

Poussez les modifications sur votre dépôt git (**git push -u origin master**). Saisissez le nom d'utilisateur puis la valeur du token généré.

Vous pouvez actualiser votre repository sur github pour voir son contenu.

## **Travail demandé :**

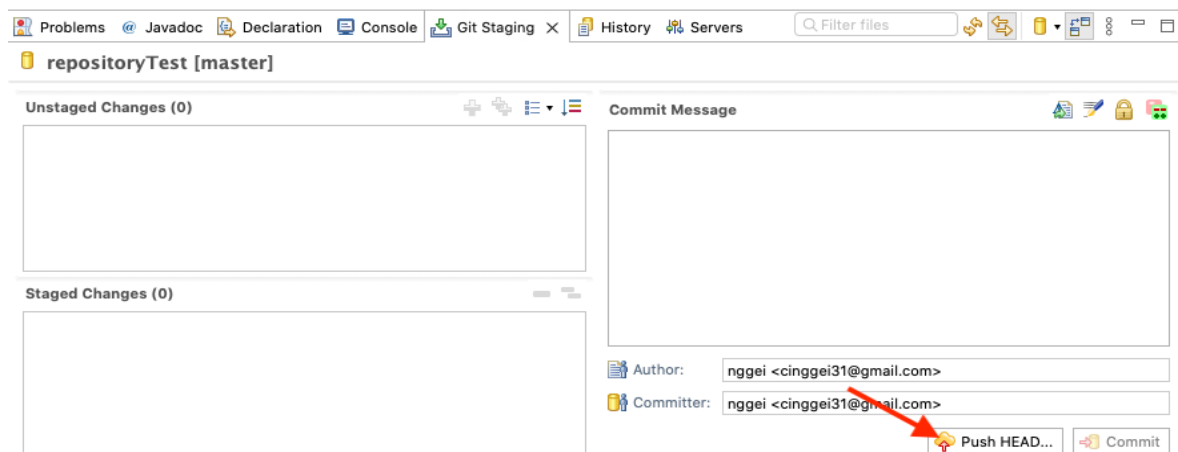
En appliquant ce que vous venez de voir, on vous demande d'effectuer des modifications de votre projet. Poussez vos modifications sur le dépôt distant.

### **3. Github et Eclipse :**

Si vous préférez utiliser Git avec Eclipse, il faut associer votre dépôt distant à Eclipse.

Retournez à Eclipse et reprenez le projet précédemment créé et versionné en local avec Git.

Aller dans *Git Staging*. Il suffit de cliquer sur « Push HEAD »



Comme votre dépôt local n'a pas été encore associé à votre dépôt Github, une fenêtre s'affiche. Vous allez configurer votre dépôt. Dans URI, saisissez l'URL de votre dépôt distant. Saisissez aussi votre nom d'utilisateur ainsi que le token généré.

The screenshot shows the 'Push Branch master' dialog box with the 'Destination Git Repository' tab selected. The dialog is titled 'Push Branch master' and has a subtitle 'Enter the location of the destination repository.' The fields are as follows:

- Remote name:
- Location:
  - URI:    - Host:
  - Repository path:
- Connection:
  - Protocol:
  - Port:
- Authentication:
  - User:
  - Password:
  - ☒ Store in Secure Store

At the bottom, there are buttons: '?', '< Back', 'Preview >', 'Cancel', and 'Push'.

Cliquez sur *Preview*. Une fenêtre vous demandant de confirmer le push s'affiche. Cliquez sur *Push*.

The screenshot shows the 'Push Confirmation' dialog box. The title is 'Push Confirmation' and the subtitle is 'Confirm following expected push result.' The dialog displays the following information:

- Branch: master → master [new branch]
- Message Details:
  - Repository: <https://github.com/nggei/Test.git>
- Options:
  - ☐ Cancel push if result would be different than above because of changes on remote
  - ☐ Show dialog with result only when it is different from the confirmed result above

At the bottom, there are buttons: '?', '< Back', 'Preview >', 'Cancel', and 'Push'.

Vous pouvez vérifier votre dépôt distant. Les modifications sont poussées.