

# Guide étudiant : Mise en place de l'environnement informatique pour la formation en Intelligence Artificielle - Datalab

---

## Sommaire

### DataLab Partie 1 : Sauvegarde et partage des TP sur "GitLab IA"

- [1. Présentation de l'environnement Datalab](#)
- [2. Sauvegarde des TP sur GitLab IA](#)
  - [Présentation de GIT](#)
  - [Connexion SSO sur GitLab IA](#)
  - [Pré-requis : compte GitLab ajouté au groupe des étudiants](#)
  - [Récupération du TP enseignant : Fork Git](#)
  - [Génération d'un Token d'authentification sur Gitlab](#)
- [3. Partage du TP](#)
  - [Partage aux autres étudiants du groupe](#)
  - [Partage à l'enseignant](#)

### 1. Présentation de l'environnement Datalab

L'environnement Datalab permet d'assurer la réalisation des TP en IA.

Cet environnement est composé de plusieurs briques technologiques :

- un système de versionnage des TP via le serveur GitLab "gitlab-ia"
  - Pour les supports et les rendus de TP, pourrait également servir de vitrine pour les projets étudiants (accessible depuis l'extérieur)
- un catalogue de données ouvertes via RUDI (Rennes Urban Data Interface)
  - Pour les datasets volumineux
- un environnement d'exécution CPU/GPU cloud via Thalès S3NS / Google Cloud
  - Les TP seront effectués dans cet environnement
- un espace de stockage de travail via Google Cloud Storage
  - Disponible pour stocker les données durant les TP si besoin, voir entre les TP

Ce guide a pour but de présenter les différentes étapes requises pour créer un TP IA.

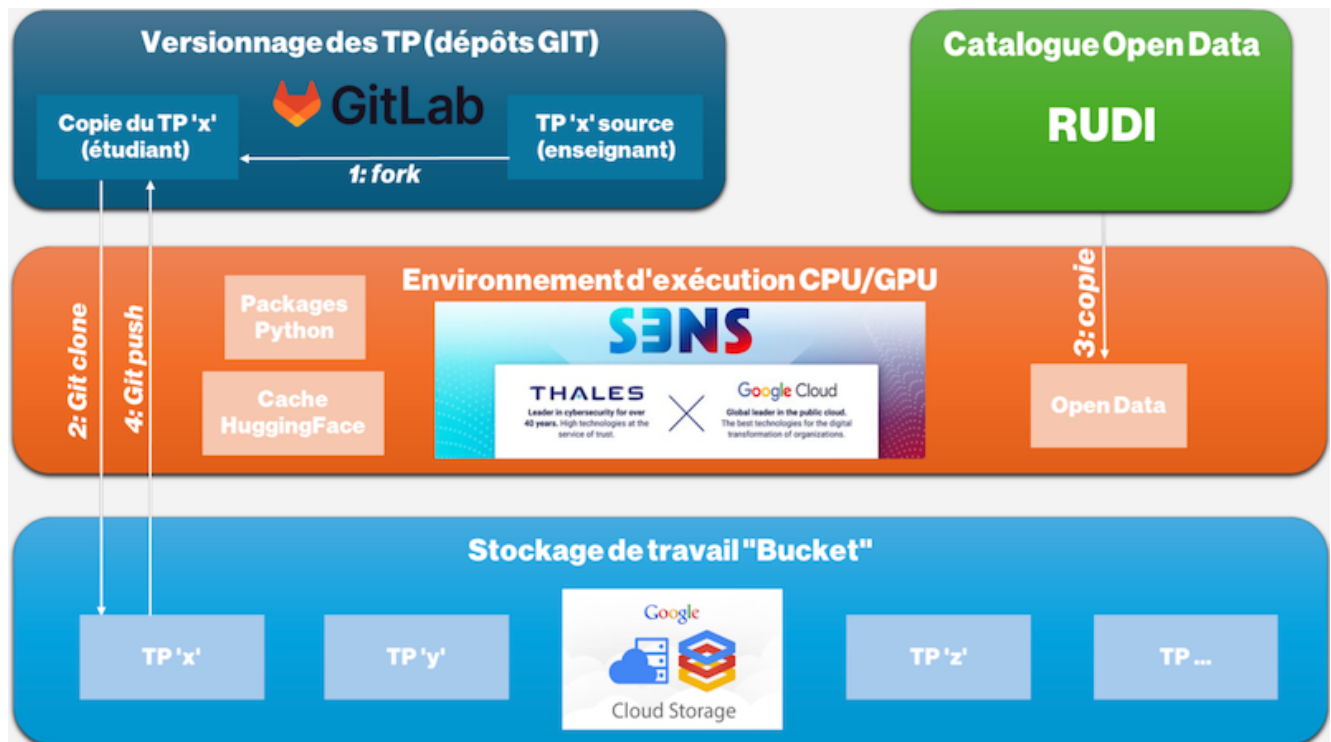


Schéma d'architecture du Datalab IA

## 2. Sauvegarde des TP sur GitLab IA

Chaque TP est sauvegardé sur le GitLab "gitlab-ia" par l'enseignant.

Tout d'abord commençons par rappeler ce qu'est "GIT".

### Présentation de GIT

GIT est un logiciel de gestion de versions : l'idée est de pouvoir sauvegarder l'historique de chacun des changements dans un ensemble de fichiers (code, notebook pour un TP,...).



*Git logo (CC 3.0 license), source <https://commons.wikimedia.org/wiki/File:Git-logo.svg>*

L'architecture de GIT est dite distribuée : on peut ainsi sauvegarder les changements localement sur une machine mais également synchroniser l'historique avec un serveur GIT distant.

C'est ce qui permet à plusieurs développeurs de travailler sur un même projet en parallèle avec des opérations de gestion des conflits si deux développeurs ont modifié un même segment de code.

GitLab est une plateforme web Open Source de développement complète qui s'appuie sur GIT mais propose également d'autres fonctionnalités :

- Gestion des problèmes : "Issues"
- Intégration continue : "build", "test", "deploy"
- Déploiement
- Analyse de la qualité du code
- Stockage de packages logiciels
- ...

Pour le projet DataLab, nous avons mis en place notre propre serveur GitLab et nous allons voir comment l'utiliser dans la prochaine partie.

---

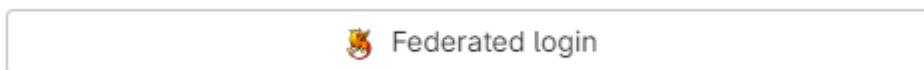
## Connexion SSO sur GitLab IA

---

Maintenant que le système de gestion de versions GIT a été rapidement présenté, voyons comment initialiser un espace GIT pour le TP.

Pour commencer, il faut se connecter avec son compte Sésame sur le serveur Git du DataLab IA disponible à l'adresse <https://gitlab-ia.univ-rennes.fr>

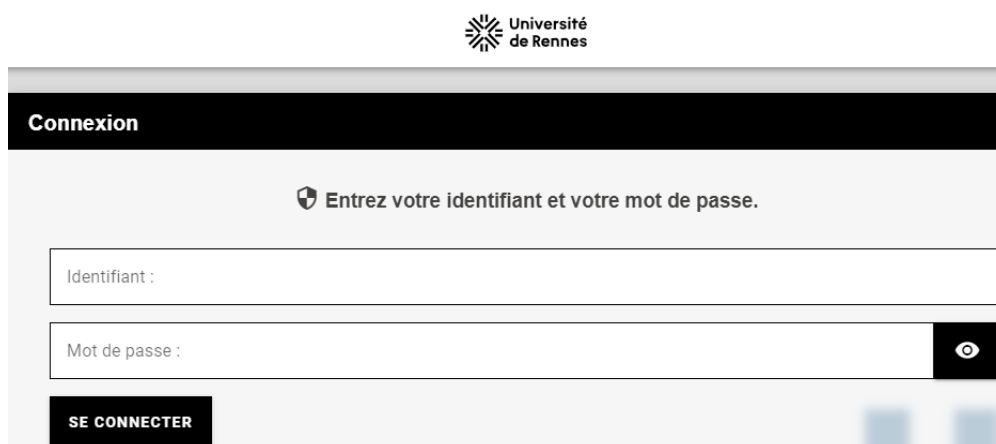
Puis il faut choisir de se connecter en utilisant la méthode "Federated login"



---

*capture login gitlab, choix de la méthode "Federated login"*

Sur l'écran suivant, il suffit de rentrer son login et mot de passe Sésame puis de cliquer sur "se connecter".

A screenshot of a login page for the University of Rennes. At the top is the university's logo and name. Below is a black header with the word "Connexion". The main area has a light gray background with the text "Entrez votre identifiant et votre mot de passe." followed by two input fields: "Identifiant :" and "Mot de passe :". The password field has a toggle icon (an eye) to its right. At the bottom left is a black button with the text "SE CONNECTER".

---

*capture login gitlab, login et mot de passe Sésame*

S'il s'agit de votre première connexion au GitLab IA, vous ne ferez partie d'aucun groupe par défaut.

Pour que votre compte soit correctement configuré, il faudra alors envoyer un mail à [support-datalab@univ-rennes.fr](mailto:support-datalab@univ-rennes.fr) avec pour titre "gitlab-activation" pour informer de votre première connexion au service.

**NOTE:** Cette étape est importante pour avoir les droits nécessaires à la création et au partage de dépôts GIT.

---

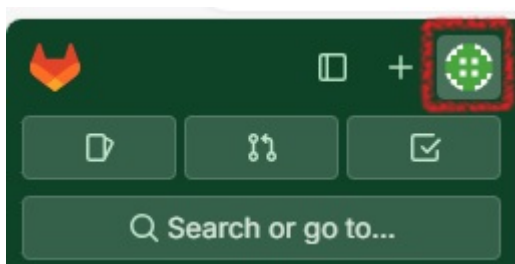
## Pré-requis : compte GitLab ajouté au groupe des étudiants

---

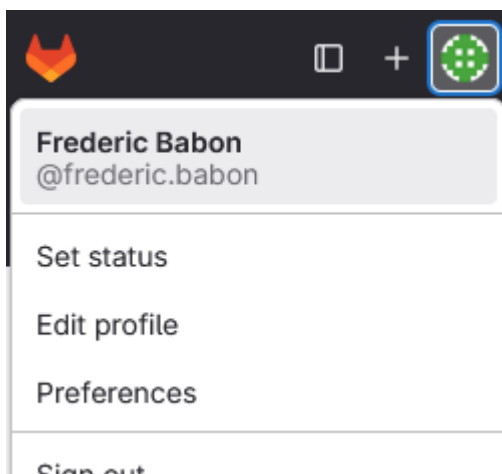
Pour parcourir les TP mis à dispositions par les enseignants, il convient de vérifier que votre compte a bien été ajouté au groupe des étudiants.

Pour cela, il faut se connecter sur <https://gitlab-ia.univ-rennes.fr>

Puis cliquer sur votre icone de profil en haut à gauche et choisir le premier item (informations du profil) pour voir plus d'informations.

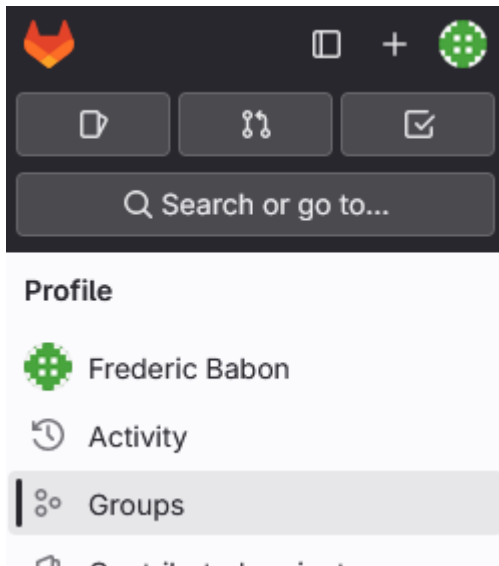


*Icône de profil Gitlab*



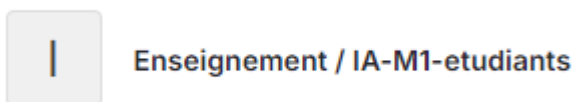
*Lien vers informations du profil*

Il suffit alors de cliquer sur Groups pour voir la liste des groupes auxquels notre compte est rattaché.



*Bouton "Groups"*

Vous devriez alors voir dans la liste des groupes du compte un ou plusieurs groupes comme :  
"Enseignement / <Formation>-etudiants"



*Exemple de groupe etudiants*

**NOTE:** Chaque étudiant fait également parti du sous-groupe "Enseignement/<Formation>-etudiants/<Formation>-etudiants-et-enseignants", c'est normal.

Ce qui valide que votre compte est bien reconnu comme étudiant pour cette formation.

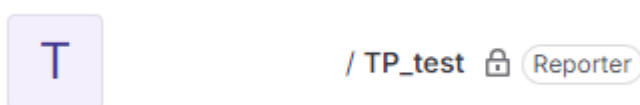
## Récupération du TP enseignant : Fork Git

Avant de commencer un TP, il faut tout d'abord faire une copie ("fork") du projet GIT du TP créé par l'enseignant.

Une fois connecté sur GitLab, vous devriez voir le TP partagé apparaître dans la page d'accueil.

Lien direct vers le projet : [https://gitlab-ia.univ-rennes.fr/frederic.babon/TP\\_test](https://gitlab-ia.univ-rennes.fr/frederic.babon/TP_test)

Le projet est partagé avec un rôle reporter, ce qui ne vous donne que des droits de consultation.



*TP partagé avec rôle reporter*

Cliquez sur ce projet pour l'ouvrir, puis choisir le bouton en haut à droite "Fork"



---

*Création d'un nouveau projet à partir de ce TP partagé ("fork")*

L'écran suivant permet de définir les propriétés élémentaires du nouveau projet :

- nom du projet : bonne pratique de nommage "TP\_" + nom du TP
- namespace : choisir son namespace personnel "<prenom.nom>" dans la liste
- le "slug" : correspond à la dernière partie de l'url GIT, il vaut mieux réutiliser le nom du projet pour éviter toute confusion
- une description du projet

A screenshot of the 'Create new fork' form in GitLab. The form has a black header bar. Below it, there is a 'Project name' field with a cursor. A red error message 'Please fill out this field.' is displayed below the name field. The 'Project URL' section contains a text field with 'https://gitlab-ia-test.univ-rennes.fr/' and a dropdown menu labeled 'Select a namespace'. To the right is a 'Project slug' field. Below these fields is a link: 'Want to organize several dependent projects under the same namespace? Create a group'. At the bottom is a 'Project description (optional)' text area.

---

*Correction du "Project slug" pour correspondre au nom de projet*

Une fois les champs renseignés, il reste à appuyer sur "Fork project" pour créer le projet.



### Fork project

A fork is a copy of a project. Forking a repository allows you to make changes without affecting the original project.

#### Project name

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

#### Project URL

#### Project slug

Want to organize several dependent projects under the same namespace? [Create a group](#)

#### Project description (optional)

#### Branches to include

- ☒ All branches  
☐ Only the default branch `main`


#### Visibility level [?](#)

- ☒ **Private**  
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.
- ☐ **Internal**  
The project can be accessed by any logged in user.
- ☐ **Public**  
The project can be accessed without any authentication.

---

### Définition des propriétés du projet GIT

Le projet est alors bien créé dans Gitlab !

 The project was successfully forked.



---

### Projet créé dans Gitlab

Si vous retournez sur la page d'accueil de Gitlab, vous devriez voir, en plus du TP partagé par l'enseignant, votre projet (copie du TP) sur lequel vous avez tous les droits (owner)



---

### Projet visible dans Gitlab

---

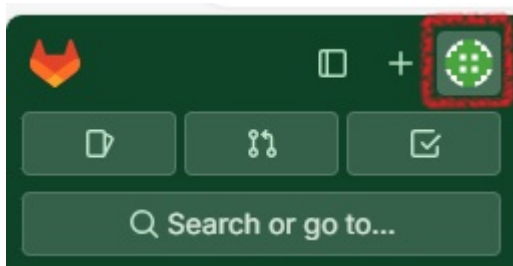
## Génération d'un Token d'authentification sur Gitlab

---

Pour pouvoir cloner le projet nouvellement créé, il va falloir tout d'abord générer un "token" d'authentification sur Gitlab.

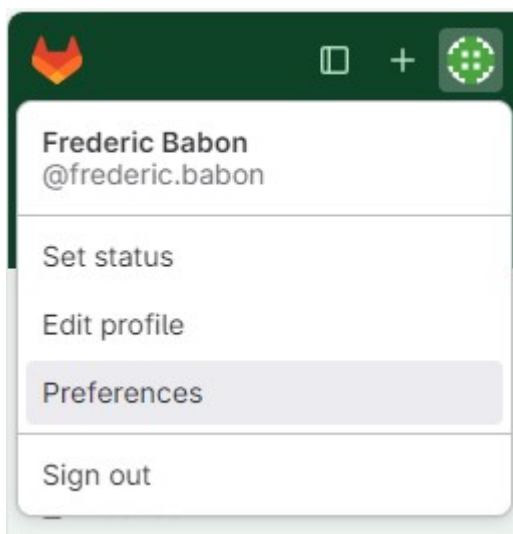
Ce Token nous permettra ensuite de récupérer le projet depuis une application extérieure (comme Google Cloud)

Le token peut se récupérer depuis notre profil Gitlab, pour accéder au profil cliquer sur l'icône de profil en haut à gauche



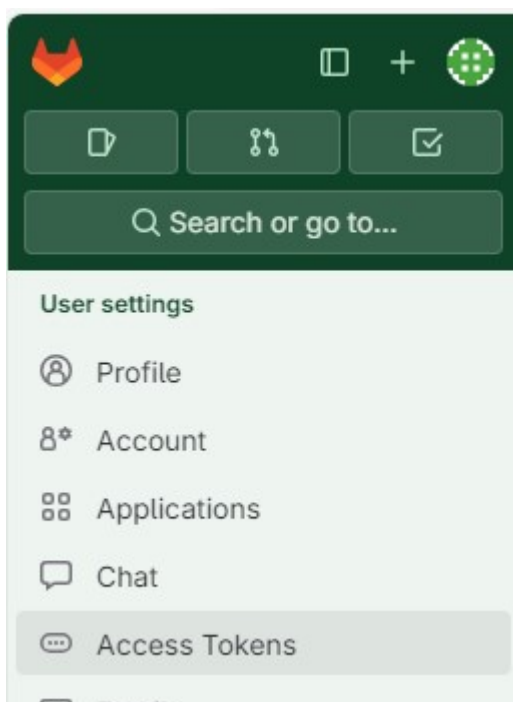
*Icone de profil Gitlab*

Cliquez alors sur "Preferences"



*Preferences Gitlab*

Puis "Access Tokens" pour accéder à la section correspondante





---

*Preferences Gitlab*

Dans l'écran suivant de gestion des tokens, il faut choisir "Add new token" pour pouvoir créer notre token.

Add new token

---

*Add new token*

On nous propose alors de choisir un nom de token d'authentification, une date d'expiration de ce token et d'activer un certain nombre de permissions.

**NOTE:** Pour la date d'expiration, une bonne pratique est de limiter autant que possible la durée de vie du token pour des raisons de sécurité

Pour les permissions, "read\_repository" and "write\_repository" doivent être activés pour pouvoir lire et écrire dans les dépôts Gitlab.

Une fois les paramètres du token configurés, cliquez sur "Create personal access token" pour obtenir le token !

Add a personal access token

Token name

my\_token\_name

For example, the application using the token or the purpose of the token.

Expiration date

2024-06-23

Select scopes

Scopes set the permission levels granted to the token. [Learn more.](#)

☐ api

Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.

☐ read\_api

Grants read access to the API, including all groups and projects, the container registry, and the package registry.

☐ read\_user

Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.

☐ create\_runner

Grants create access to the runners.

☐ k8s\_proxy

Grants permission to perform Kubernetes API calls using the agent for Kubernetes.

☒ read\_repository

Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.

☒ write\_repository

Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

☐ read\_registry

Grants read-only access to container registry images on private projects.

☐ write\_registry

Grants write access to container registry images on private projects. You need both read and write access to push images.

☐ ai\_features

Grants access to GitLab Duo related API endpoints.

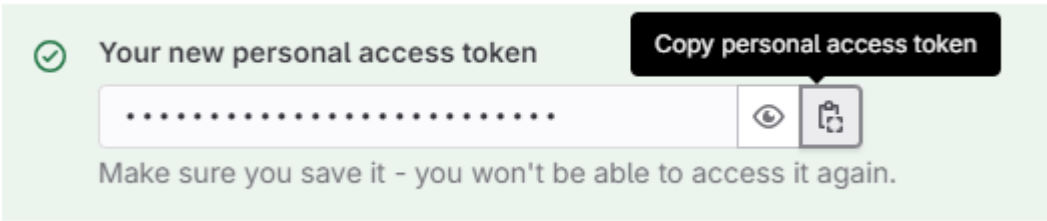
Create personal access token

Cancel

---

*Add a Personal access token*

Vous verrez alors apparaitre sur l'écran suivant une section "Your new personal access token".



section "Your new personal access token"

Il faut absolument copier le "personal access token" généré car il ne sera pas possible d'y accéder autrement !

**NOTE:** Si vous perdez le token (ou s'il expire), il est possible de régénérer le token avec les mêmes options en cliquant sur les 3 petits points et choisir "rotate"

Name	Status	Scopes	Usage ?	Lifetime	
august	Active	write_virtual_registry, write_repository, api	Last used: Never	in 2 weeks Sep 02, 2025	<div><div></div><div>Rotate</div><div>Revoke</div></div>

Feed token

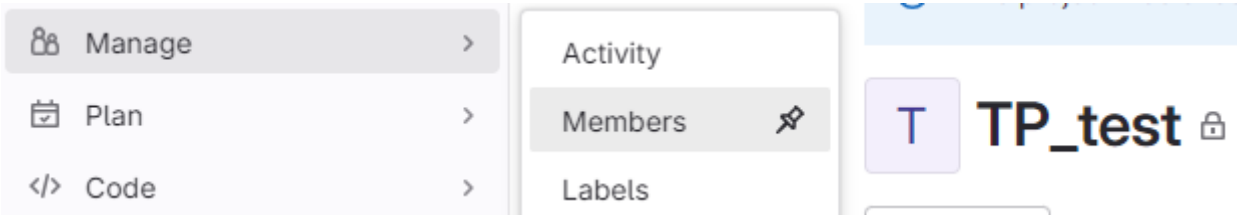
section "Your new personal access token"

**NOTE:** Une bonne pratique pour sauvegarder ce token en sécurité peut être d'utiliser un gestionnaire de mot de passe comme "Keepass" (disponible sur les machines de TP)

3. Partage du TP

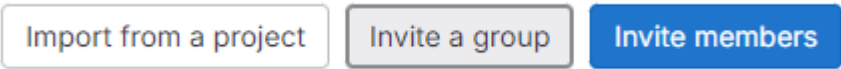
Partage aux autres étudiants du groupe

Pour partager un projet aux autres étudiants de votre groupe, il faut tout d'abord sélectionner le projet depuis la page d'accueil de <https://gitlab-ia.univ-rennes.fr> Puis choisir dans le menu de gauche Manage > Members



Menu "Manage>Members"

Cliquez sur "Invite members"

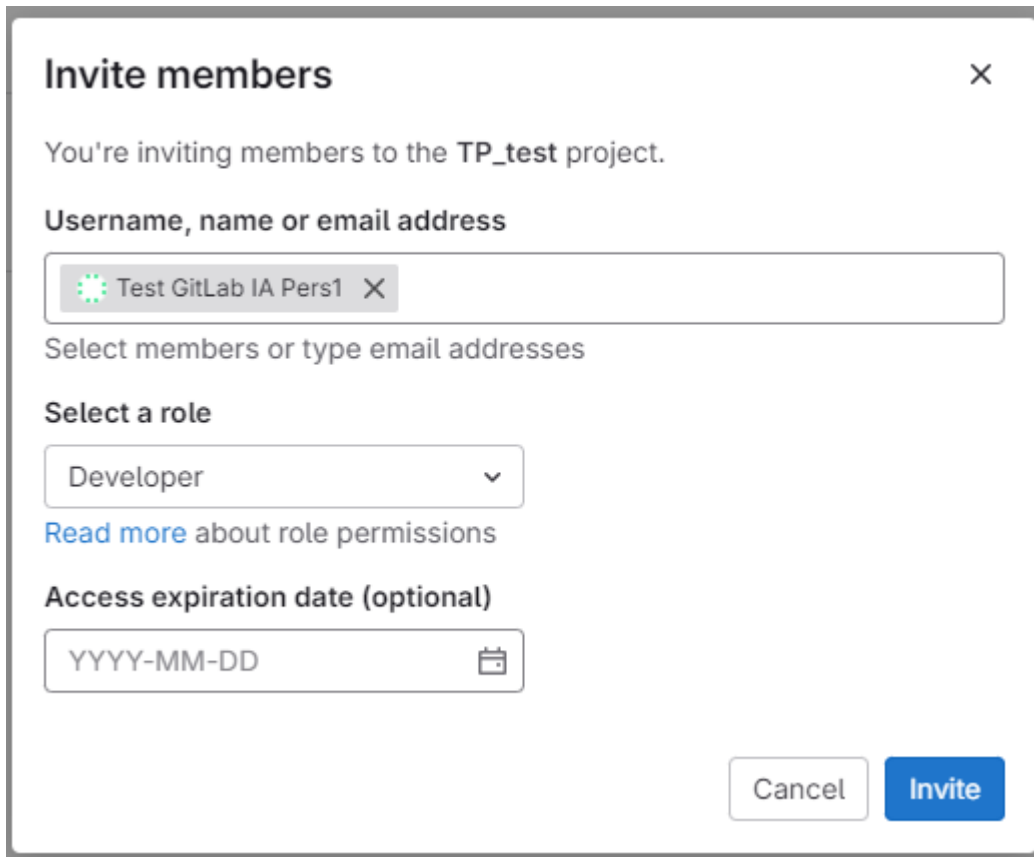


Bouton "Invite members"

Puis dans le pop-up qui s'affiche, cliquez dans le champ vide sous "Username, name or email address" pour sélectionner les étudiants visés par le partage.

Pour "Select a role" choisir "Developer"

Et il n'est pas forcément nécessaire de choisir une date d'expiration.



**Invite members** [X]

You're inviting members to the **TP\_test** project.

**Username, name or email address**

Test GitLab IA Pers1 [X]

Select members or type email addresses

**Select a role**

Developer [v]

[Read more](#) about role permissions

**Access expiration date (optional)**

YYYY-MM-DD [calendar icon]

Cancel Invite

*Selection des membres invités et du rôle*

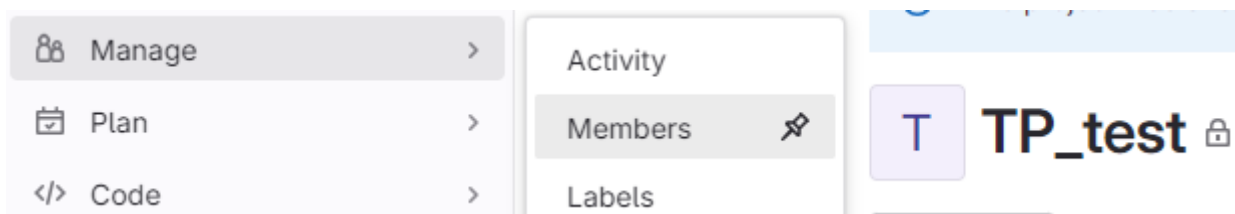
Le projet est maintenant partagé aux autres étudiants.

---

## Partage à l'enseignant

---

Pour partager un projet à un enseignant, il faut procéder de la même manière. Une fois le projet ouvert, choisir dans le menu de gauche Manage > Members



*Menu "Manage>Members"*

Cliquez sur "Invite members"

Import from a project

Invite a group

Invite members

---

*Bouton "Invite members"*

Puis dans le pop-up qui s'affiche, cliquez dans le champ vide sous "Username, name or email address" pour sélectionner le(s) enseignant(s).

Pour "Select a role" choisir "reporter"

Et il n'est pas nécessaire de choisir une date d'expiration.

Le projet est maintenant partagé à l'enseignant.

Félicitations, la partie 1 du guide DataLab est terminée !