

# GITLAB CTOM

NG version du 06/07/2022

## Table des matières

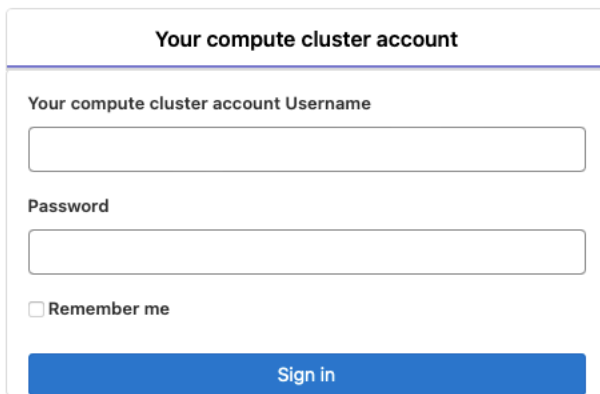
<b>1. Accès au serveur GITLAB.....</b>	<b>2</b>
<b>2. Liste des projets.....</b>	<b>2</b>
<b>2.1. Cloner un projet dans sur le poste local.....</b>	<b>3</b>
2.1.1. Récupérez l'adresse du projet.....	3
2.1.2. Clonage en ligne de commande .....	3
2.1.2.1. Clonage HTTPS en ligne de commandes .....	3
2.1.2.2. Clonage SSH en ligne de commandes .....	4
2.1.2.3. Commandes GIT utiles .....	4
2.1.3. Récupérer le projet sous Eclipse IDE .....	5
2.1.3.1. Importer le projet GIT dans Eclipse : .....	5
2.1.3.2. Créer un projet Eclipse basé sur un dépôt local GIT .....	8
2.1.3.3. Utilisation de GIT sous Eclipse .....	10
<b>3. Particularités pour le projet HuLiS.....</b>	<b>10</b>

Un serveur GITLAB a été mis en place pour gérer les codes sources de l'équipe CTOM

## 1. Accès au serveur GITLAB

L'interface Web du serveur GITLAB est accessible à l'adresse <https://alta.ism2.dcs.univ-amu.fr/gitlab/dryhou36/hulisHTML5>

Connectez-vous avec l'identifiant et le mot de passe que vous utilisez habituellement pour le cluster de calcul Les Cités Obscures de l'équipe CTOM

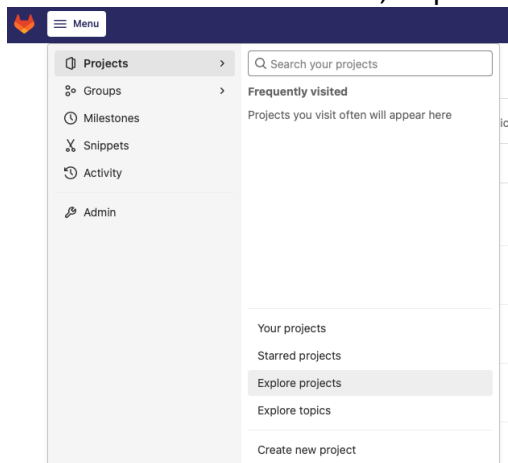


The screenshot shows a login form with the following elements:

- Title: **Your compute cluster account**
- Label: **Your compute cluster account Username**
- Input field for Username
- Label: **Password**
- Input field for Password
- Checkbox:  Remember me
- Button: **Sign in**

## 2. Liste des projets

Dans l'interface Web GITLAB, cliquer sur le bouton *Menu / Projects / Explore Projects*



Les projets HuliS ont été organisés selon leur nature : Java SWING, GWT, HTML5.

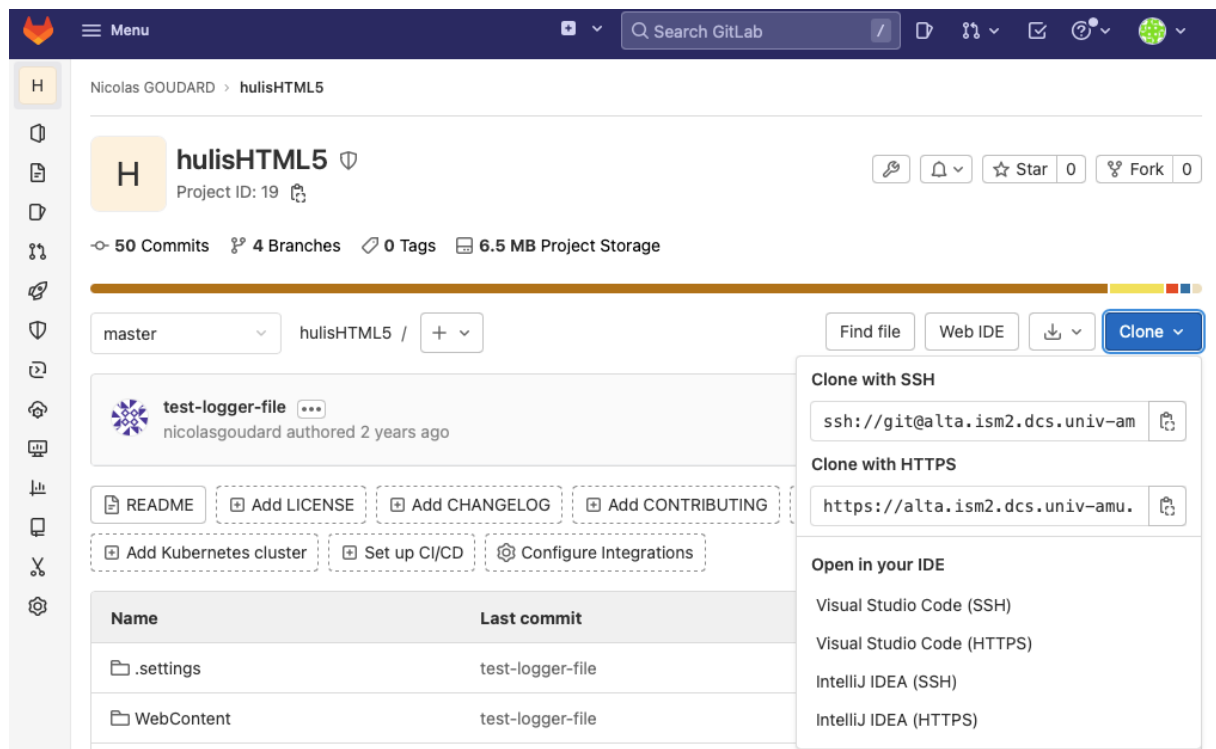
- HulisSWING : projet SWING, le code source de la version classique en Java de HuliS
- HulisGWT : projet GWT : version mobile, bientôt remplacée par HTML5
- hulisHTML5. : projet HTML5 en développement
- hulis\_dev : Règles de développements, tâches à effectuer, liste des bugs à corriger

## 2.1. Cloner un projet dans sur le poste local

Dans cette notice on proposera de récupérer le projet en lignes de commande ou avec Eclipse

### 2.1.1. Récupérez l'adresse du projet

Pour récupérer le projet souhaité avec GIT sur votre poste, connectez-vous sur l'interface Web GITLAB, cliquez sur le nom projet à récupérer. Cliquer ensuite sur le bouton *Clone* pour copier son adresse. Le plus simple est de copier l'adresse en HTTPS



### 2.1.2. Clonage en ligne de commande

#### 2.1.2.1. Clonage HTTPS en ligne de commandes

Ouvrez un terminal, aller dans un dossier de développement puis tapez la commande :

```
$ git clone https://alta.ism2.dcs.univ-amu.fr/gitlab/pathto/project.git
```

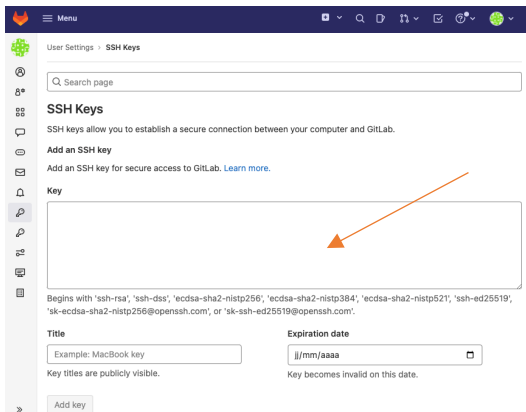
Puis entrez votre identifiant GITLAB



### 2.1.2.2. Clonage SSH en ligne de commandes

Dans l'interface Web GITLAB, vous devez autoriser votre poste de travail à accéder en SSH aux projets.

Pour cela, allez dans votre *profil / Edit profile / SSH Keys* et ajoutez votre clé publique SSH qui se trouve généralement dans le fichier `~/.ssh/id_rsa.pub` sur votre poste de travail. Copiez tout le contenu du fichier.



Le serveur GITLAB se trouve derrière un HAProxy ce qui implique une configuration particulière : ajouter la configuration de proxy suivante dans le fichier `~/.ssh/config`

```
Host gitserver
  User git
  HostName webservices.ism2.dcs.univ-amu.fr
  ProxyCommand openssl s_client -quiet -connect alta.ism2.dcs.univ-
  amu.fr:22022 -servername bk_ssh_gitlab
```

Récupérez l'adresse du projet et adaptez la dans la commande « git clone » en remplaçant « `git@alta.ism2.dcs.univ-amu.fr:22022` » par l'hôte virtuel « `gitserver` » défini précédemment dans le fichier `~/.ssh/config`. Ainsi la commande pour cloner le projet hulisHTML5 sera :

```
$ git clone ssh://gitserver/dryhou36/hulisHTML5
```

Si la commande git renvoie l'alerte suivante : `WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!` effacez la clé obsolète dans le fichier `~/.ssh/known_hosts` en exécutant la commande :

```
$ ssh-keygen -R webservices.ism2.dcs.univ-amu.fr
```

### 2.1.2.3. Commandes GIT utiles

Cloner le projet : `git clone ssh://git@ip:port/repo/gitfile.git`

Lister toutes les branches : `git branch -a`

Changer la branche courante : `git checkout nom_de_la_branche`

Créer une branche locale : `git checkout -b nom_de_la_branche`

Envoyer une branche sur le serveur : `git push origin nom_de_la_branche`

Récupérer les modifications faites dans la branche courante : `git pull`

Modifier la branche courant en local : `git commit -m "message" fichier1 fichier2`

Envoyer les modifications locales sur le serveur : `git push`

Documentation GITLAB sur internet : <https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html>

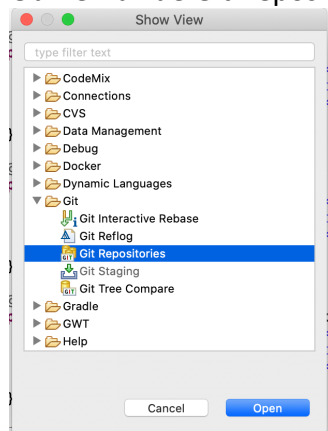
### 2.1.3. Récupérer le projet sous Eclipse IDE

Vous devez avoir installé *Eclipse IDE For Enterprise Java and Web Developers* : <https://www.eclipse.org>

Le clonage se fera en utilisant l'adresse HTTPS du projet car il semble que la version GIT de Eclipse, EGIT, ne supporte pas le clonage SSH via un HAProxy.

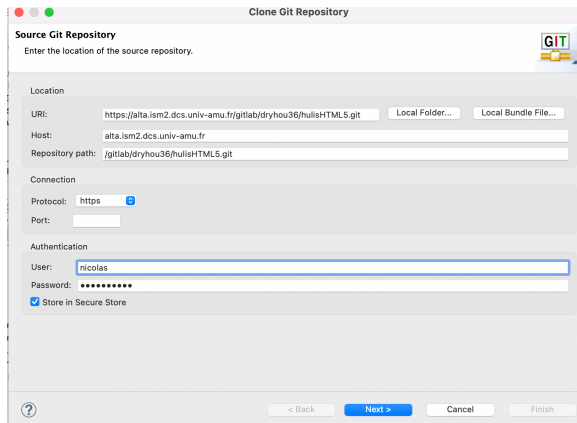
#### 2.1.3.1. Importer le projet GIT dans Eclipse :

Ouvrez la vue Git Repositories (Menu Windows / Views / Others)

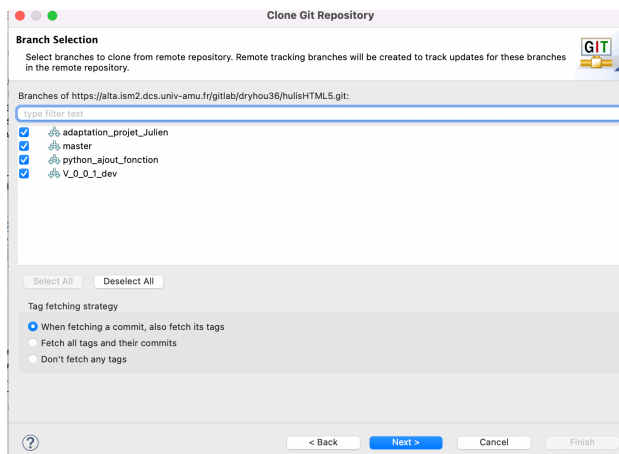


Dans la vue Git Repository, cliquez sur le bouton clone. 📁➡️

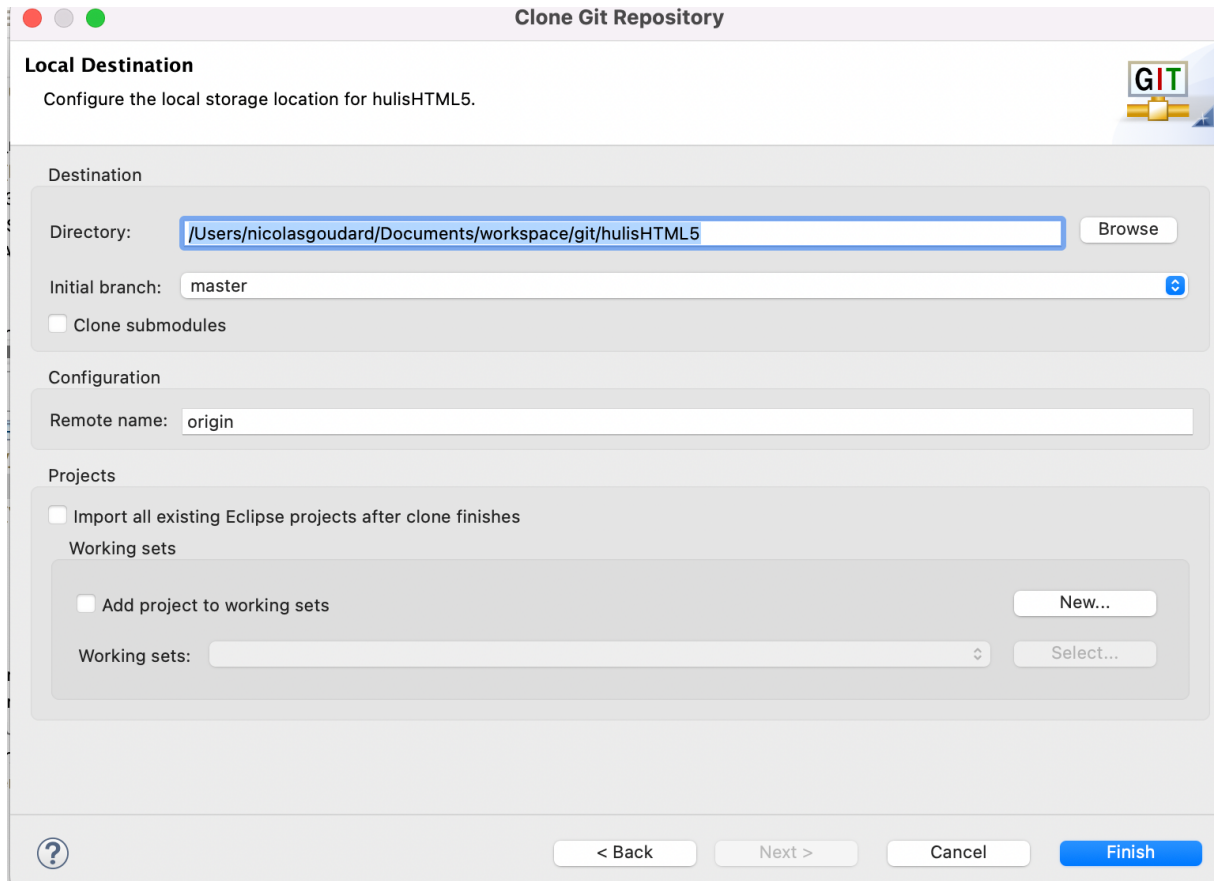
Indiquez l'URL HTTPS du projet dossier de destination, les identifiants et le mot de passe GITLAB



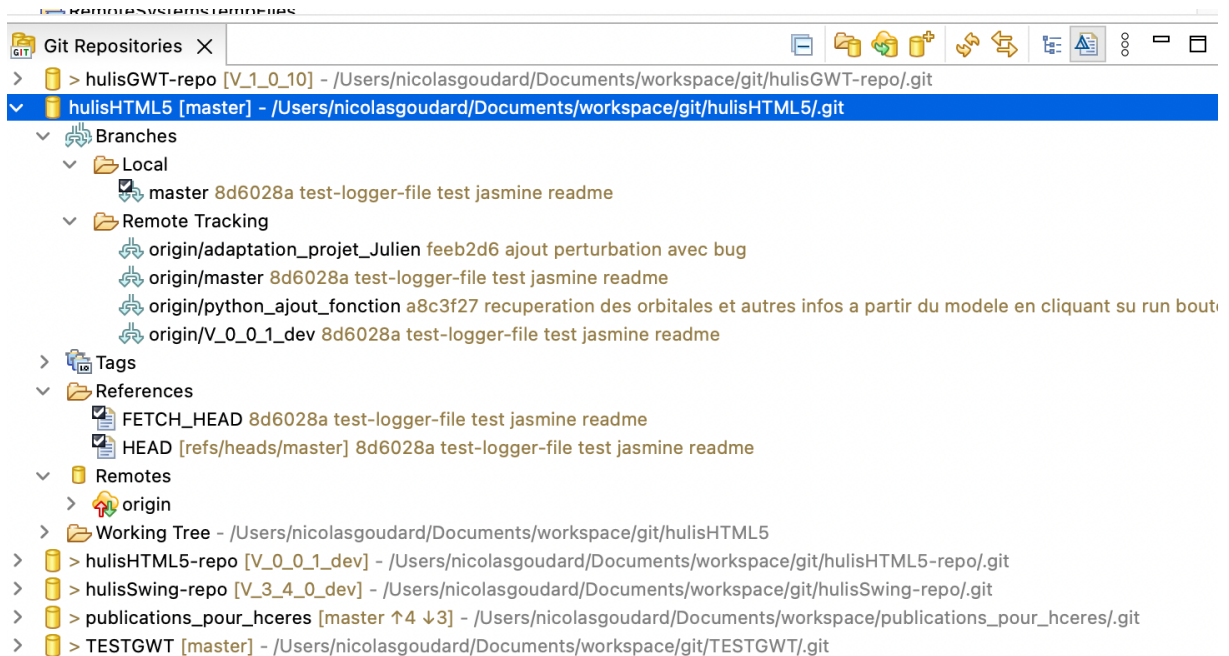
Sélectionnez toutes les branches du projet



Sélectionnez un dossier de destination local, par exemple dans votre espace de travail Eclipse et cliquez sur le bouton Finish



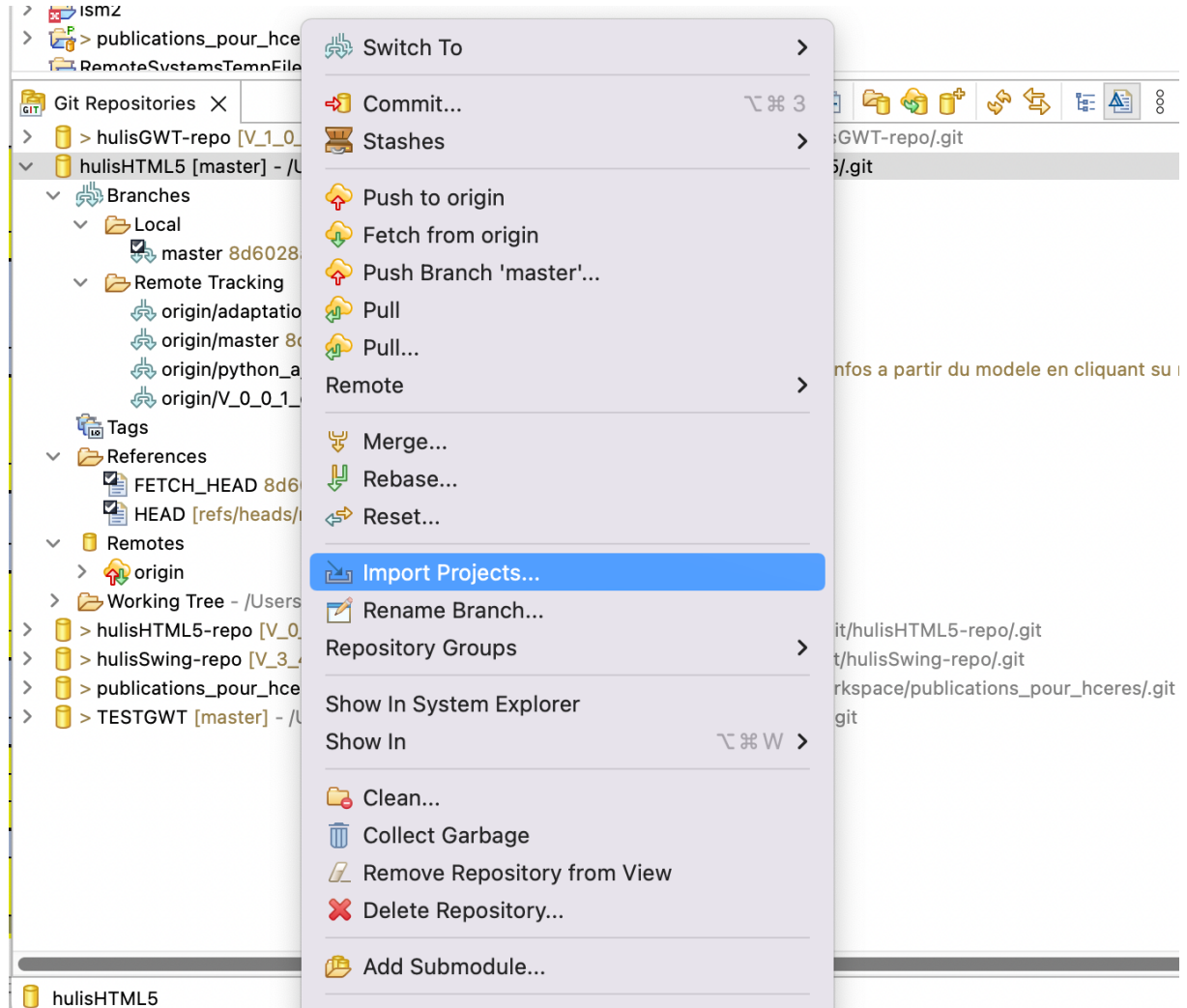
Le projet apparaît bien cloné dans la vue Git Repositories



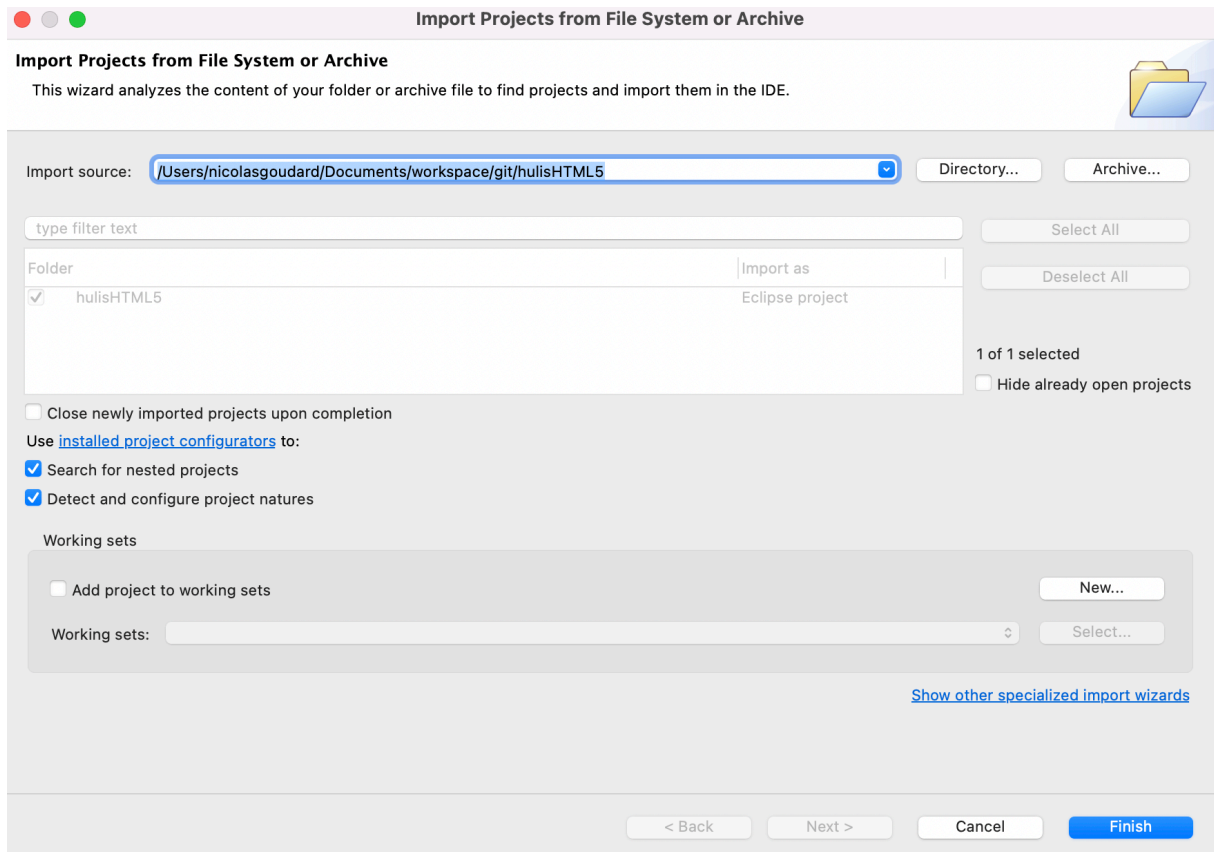
### 2.1.3.2. Créer un projet Eclipse basé sur un dépôt local GIT

Créez à présent un projet dans Eclipse basé sur le dépôt GIT local

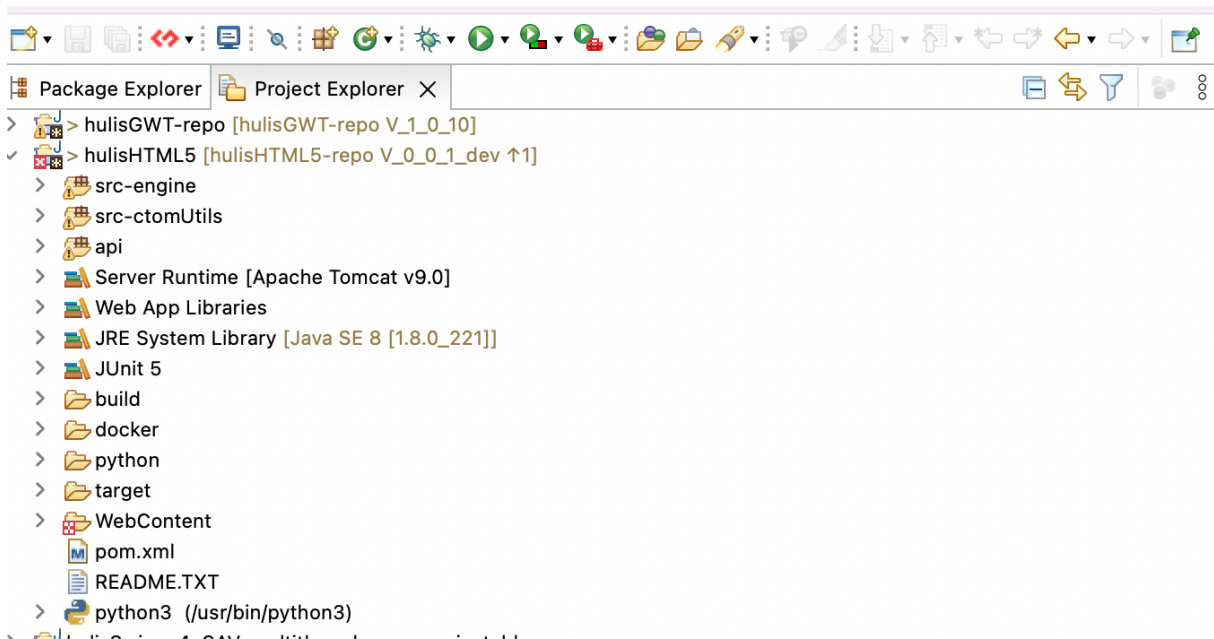
Bouton droit sur le dépôt local dans la vue *Git Repositories / Import Project / Finish*







Le projet est maintenant visible dans la vue *Project Explorer*



### 2.1.3.3. Utilisation de GIT sous Eclipse

Vous pouvez maintenant développer dans le dépôt git local en liaison avec le serveur GITLAB. Veuillez nommer les branches selon les consignes de développement indiquées dans le projet hulis\_dev sur le serveur GITLAB (voir précédemment).

Les commandes GIT se trouvent dans le menu *Team* (Bouton droit sur le projet)

**Pull** : récupère les modifications du serveur GITLAB vers le dépôt GIT local

**Add to index** : ajoute les fichiers dans le prochain commit

**Commit** : enregistre les modifications dans le dépôt GIT local. Faire Add to index sur les fichiers désirés avant

**Switch to** : changer de branche / créer une nouvelle branche

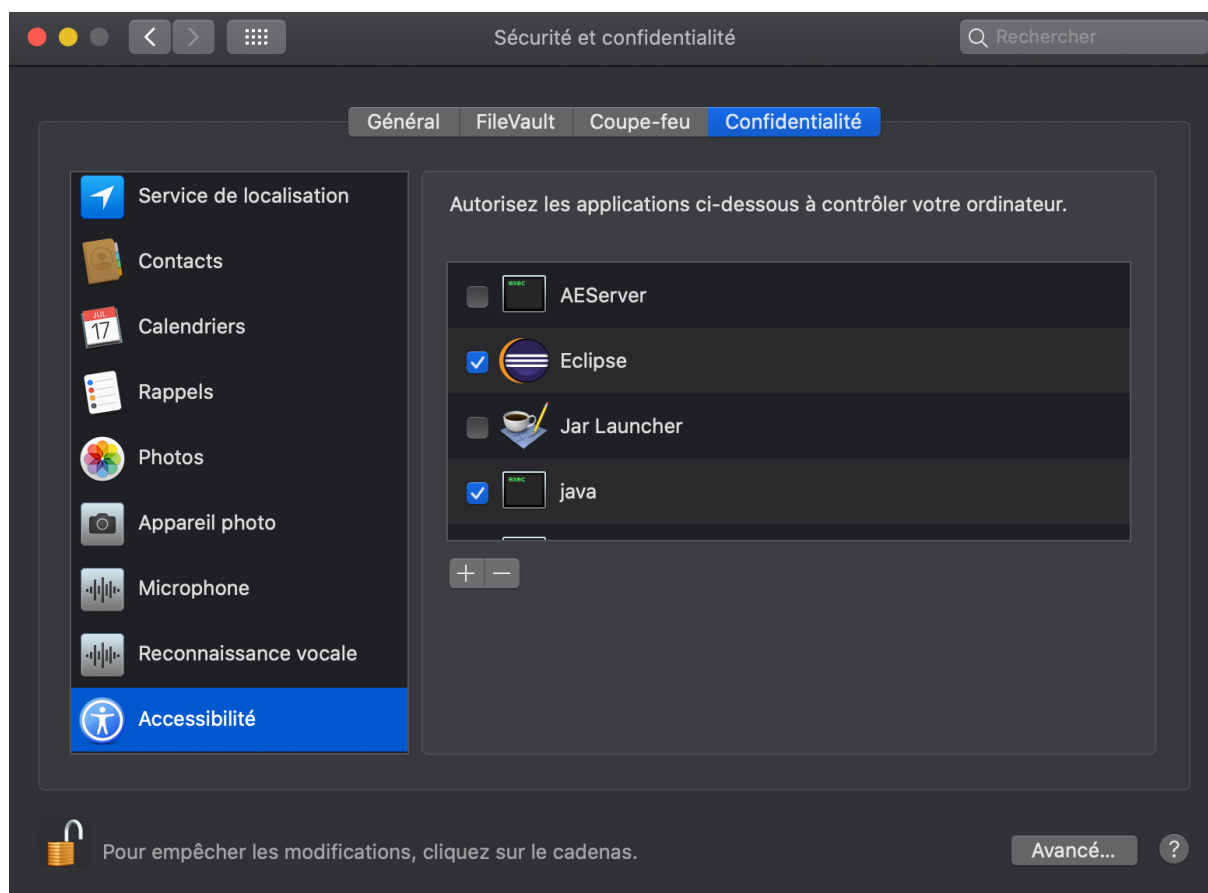
**Push** : envoie les modifications du dépôt local vers le serveur GITLAB pour les partager

Toutes les modifications effectuées sur le dépôt GIT local ne seront pas visibles par les autres développeurs tant que vous n'aurez pas effectué la commande « Push », de même pour récupérer les modifications faites par les autres vous devez faire « Pull » de temps en temps.

## 3. Particularités pour le projet HuLiS

HulisHTML5 : suivre les instructions du fichier Readme.txt à la racine du projet pour la configuration

HulisSWING : Sous MacOs / Pomme / préférences systèmes / Sécurité et Confidentialité, à autoriser Java et Eclipse à contrôler l'ordinateur, sinon le Robot du Tutorial et les tests interactifs ne fonctionneront pas



## Information sur le serveur GITLAB et la maintenance

Le serveur de développement GITLAB est hébergé au service 561 sur un NAS Synology

La lenteur d'accès peut être attribuée à la configuration insuffisante du projet : Le projet synology-gitlab recommande 4GO de RAM au lieu des 2GO actuels sur notre configuration (voir ici : <https://github.com/jboxberger/synology-gitlab>)

Le contenu du serveur GITLAB est sauvegardé tous les soir à 23h00 sur le serveur de l'iSm2

Pour tout problème, veuillez contacter [nicolas.goudard@univ-amu.fr](mailto:nicolas.goudard@univ-amu.fr)