

**DEV1 – JAVL – Laboratoires Java****TD X – Une introduction à Git sous NetBeans**

Un logiciel de gestion de versions comme GIT permet notamment de : gérer l'historique d'un projet en sauvegardant chaque version et la description de ce qu'elle apporte ; sauvegarder le projet (dans toutes ses versions) sur un serveur ; le retrouver facilement sur une autre machine.

Le but de ce TD est de vous offrir une introduction à GIT en utilisant NETBEANS. Un TD du laboratoire environnement vous permettra d'apprendre à utiliser GIT en mode commande pour aller plus loin dans sa compréhension et son utilisation.

Les codes sources et les solutions de ce TD se trouvent à l'adresse :  
<https://git.esi-bru.be/dev1/labo-java/tree/master/tdX-git/>

**Table des matières**

<b>1</b>	<b>Utiliser git avec Netbeans en local</b>	<b>2</b>
<b>2</b>	<b>Le serveur gitlab</b>	<b>5</b>

# 1 Utiliser git avec Netbeans en local

Explorons comment demander à GIT de suivre notre projet et comment créer et décrire ses différentes versions.

## Tutoriel 1

### Créer un projet sous Netbeans

Commençons par créer un projet que nous allons pouvoir faire suivre par GIT.

- ✍ Créez un nouveau projet NETBEANS appelé **TestGit**. Ce projet sera constitué :
  - ▷ d'un package **dev1.jav1** ;
  - ▷ d'une classe principale appelée **MainApp** qui affiche le numéro de version du projet.



- ✍ Vous pouvez l'exécuter pour vérifier que tout va bien.

## Tutoriel 2

### Initialisation de l'historisation

Nous allons maintenant demander à git de suivre *localement* notre projet. Pour ce faire, suivez les étapes suivantes sous Netbeans :

- ✍ Dans l'onglet **Projects** (à gauche) cliquez droit sur le projet **TestGit**.
- ✍ Choisissez l'option **Versioning**.
- ✍ Suivi de l'option **Initialize Git Repository...**



- ✍ Un pop-up vous propose de choisir le dossier qui contiendra l'historique local. Conservez le chemin proposé par défaut, il s'agit du répertoire de votre projet.

### Tutoriel 3

## Un premier commit

Imaginons que la première fonctionnalité de votre projet soit terminée : vous savez afficher la version du projet dans la console. Comme cette partie du développement est clôturée, vous souhaitez l'enregistrer dans l'historique du projet.

### Commit

Un **commit** (ou **soumission** en français) contient l'ensemble des fichiers et leur contenu qui composaient le projet à un moment donné. Il contient également une série d'informations comme une date de création, le nom du créateur et une description.

Pour effectuer ce premier commit, suivez les étapes suivantes :

- ✍ Dans l'onglet **Projects** cliquez droit sur le projet.

Attention ! Une erreur courante est de cliquer droit sur un fichier et non sur le projet. Seules les modifications de ce fichier sont committées dans ce cas, ce qui n'est généralement pas l'action souhaitée.

- ✍ Sélectionnez l'option **Git** puis l'option **Commit...**



Cette action vous renvoie vers l'écran de validation des *commits* où vous pourrez décrire ce que vous venez de développer.



On y trouve également la liste tous les fichiers qui ont été créés, modifiés ou supprimés par votre travail. Comme il s'agit d'un nouveau projet, il s'agit ici des fichiers créés et gérés par NETBEANS en plus de la classe `MainApp`.

- ✍ Écrivez le message suivant dans la partie **Commit Message**: "Création du projet et affichage du numéro de version".
- ✍ Terminez l'opération en cliquant sur le bouton **Commit**.

#### Tutoriel 4

### Nouvelle fonctionnalité... Nouveau Commit

Dans le cadre de ce projet une nouvelle fonctionnalité est demandée, l'utilisateur doit encoder son nom.

- ✍ Modifiez le programme afin qu'il demande le nom de l'utilisateur :

```
1 package be.esi.heb.user;
2
3 import java.util.Scanner;
4
5 /**
6  * @author esiProfs
7  */
8 public class MainApp {
9     public static void main(String[] args) {
10         Scanner keyboard = new Scanner(System.in);
11         String userName;
12         System.out.println("Test de la version numéro 2");
13         System.out.println("Veuillez entrer votre nom : ");
14         userName = keyboard.next();
15     }
16 }
17
```

Afin de conserver dans l'historique cette nouvelle version du projet, il faut effectuer un nouveau *commit*.

- ✍ Effectuez ce commit comme expliqué dans le tutorial précédent.



Comme vous le constatez sur l'écran de validation des commits, un seul fichier a été modifié : `MainApp.java`.

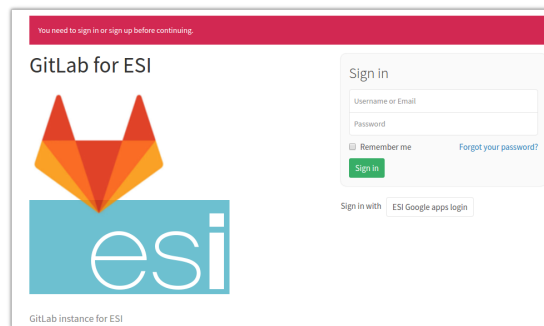
## 2 Le serveur gitlab

Pour l'instant l'historique de votre projet est enregistré dans un dossier caché de votre projet (.git). Il est possible d'utiliser git sans serveur pour conserver un historique en local mais l'utilisation d'un serveur permet de développer sur différentes machines. C'est ce que nous allons voir dans cette section.

Un **serveur git** nous permet principalement :

- ▷ de pouvoir accéder facilement à notre projet sur différentes machines ;
- ▷ de contribuer à plusieurs sur un même projet ;
- ▷ de montrer publiquement le projet à toute la communauté dès lors qu'il est public.

Nous avons installé à l'école le serveur GITLAB FOR ÉSI.



### Tutoriel 5

## S'identifier sur le serveur

Voyons comment s'identifier sur le serveur.

- ✍ Rendez-vous sur le serveur <https://git.esi-bru.be> et identifiez-vous en cliquant sur le bouton : **ESI Google apps login**.

Pour vous faciliter les opérations ultérieures, il est important de définir un mot de passe propre à ce serveur. Pour cela,

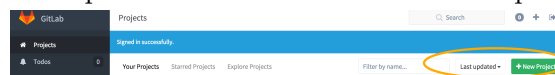
- ✍ Cliquez sur le menu en haut à droite ;
- ✍ choisissez **Profile settings** puis **Password** ;
- ✍ Cliquez alors sur **I forgot my password** et suivez la procédure.

### Tutoriel 6

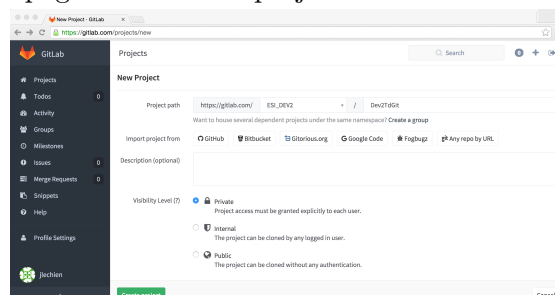
## Création d'un dépôt sur le serveur

Créon sur GITLAB un **dépôt** qui va pouvoir accueillir notre projet dans toutes ses versions.

- ✍ Sur la page d'accueil de votre compte GitLab, vous trouverez un bouton dans le coin supérieur droit vous permettant de créer des nouveaux projets. Cliquez dessus.



- ✍ Donnez à ce projet le nom : **Dev1TdGit** et confirmez la création du projet. Vous êtes redirigé vers la page d'accueil du projet.



## Premier push

Il est temps de sauver l'historique de votre projet NETBEANS sur le serveur GITLAB.

### Push

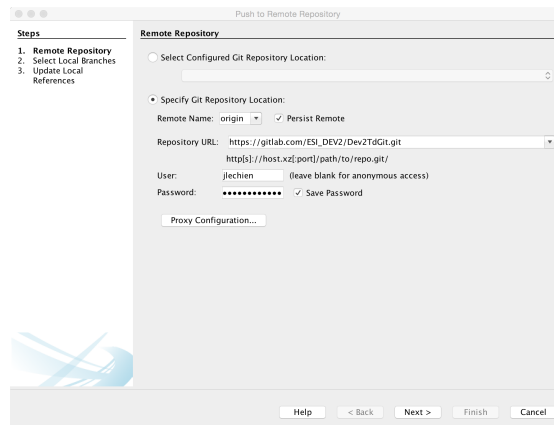
Un **push** consiste à envoyer sur le serveur tous les nouveaux commits effectués localement.

Pour effectuer ce push :

- ✍ Sur la page d'accueil de votre dépôt GitLab, récupérez l'url **https** de ce dépôt. Elle sera du type **https://git.esi-bru.be/g12345/Dev1TdGit.git**. Le plus simple est de la copier dans le presse-papier.
- ✍ Dans NetBeans :
  - ✍ Dans l'onglet **Projects** cliquez droit sur le projet.
  - ✍ Choisissez l'option **Git**, suivi de **Remote** et enfin **Push...**

Comme c'est le tout premier *push* pour ce projet, il va falloir indiquer à NETBEANS quel dépôt utilisé. Les *push* suivants seront plus immédiats.

- ✍ Dans **Remote Repository** : copiez l'url **https** de votre projet (celui que vous venez de copier) et entrez votre **username** et votre **password** (défini un peu plus tôt).



### On ne push que ce qui est commité

Attention, lors du *commit*, seuls les commits sont envoyés sur le serveur. Sauver un fichier ne suffit pas. Les modifications apportées à ce fichier ne seront pas envoyées si vous n'avez pas inclus ce fichier à un commit après avoir effectué les modifications.