Haute École Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2019 – 2020
DEV1 – JAVL

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 sauvant chaque version et la description de ce qu'elle apporte ; sauver 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

Version du document
27 octobre 2019

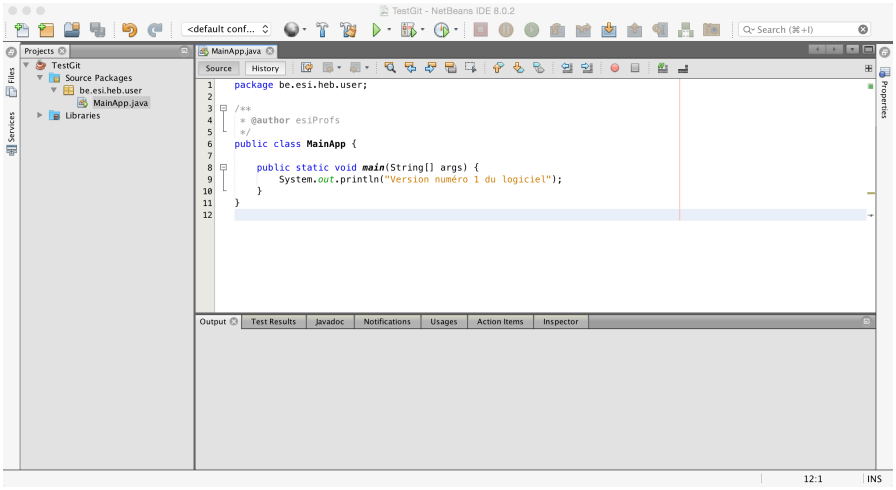
1 Utiliser git avec Netbeans en local

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.javl` ;
  - ▷ d'une classe principale appelée `MainApp` qui affiche le numéro de version du projet.

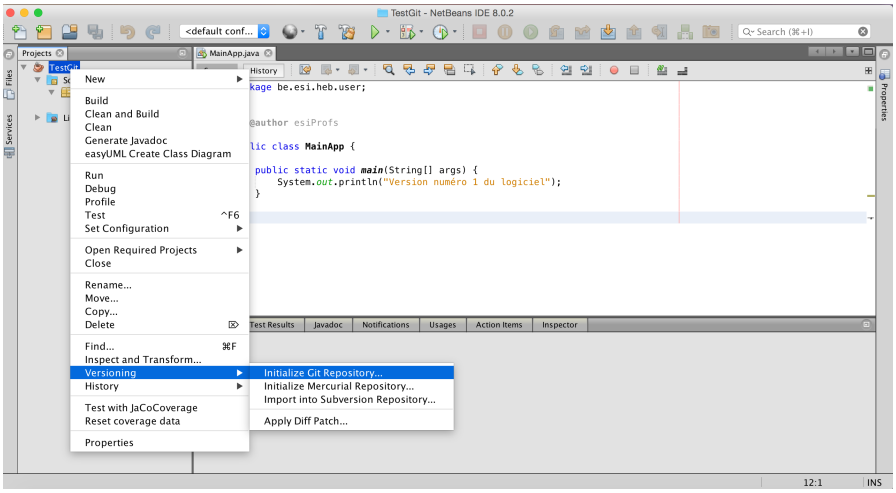


- Vous pouvez l'exécuter pour vérifier que tout va bien.

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

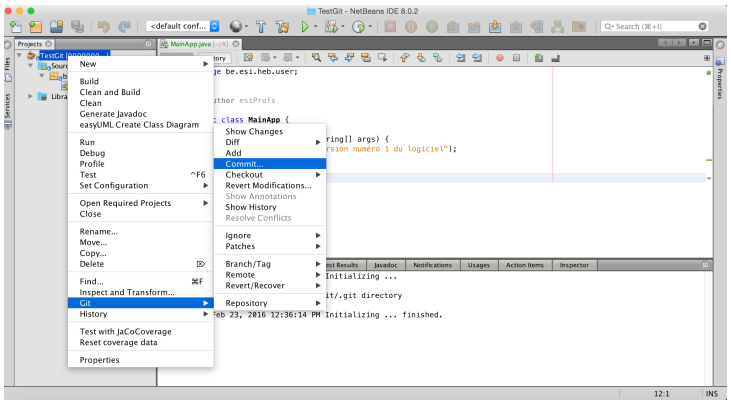
Un **commit** (ou **soumission** en français) contient l'ensemble des fichiers et leur contenu qui composaient le projet à un moment donné. Il contient également une série d'informations comme une date de création, le nom du créateur et une description.

Pour effectuer ce premier commit, suivez les étapes suivantes :

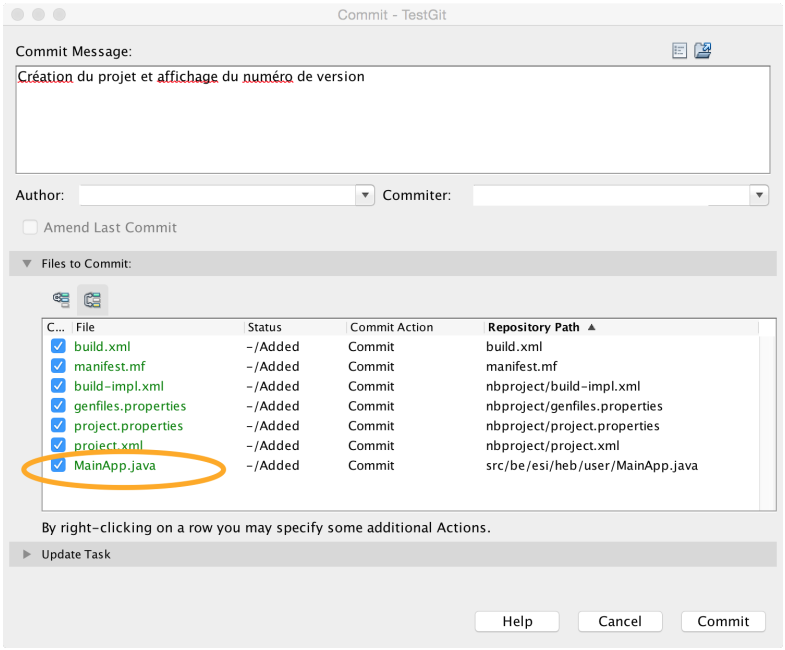
- Dans l'onglet `Projects` cliquez droit sur le projet.

> Attention ! Une erreur courante est de cliquer droit sur un fichier et non sur le projet. Seules les modifications de ce fichier sont commitées 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.



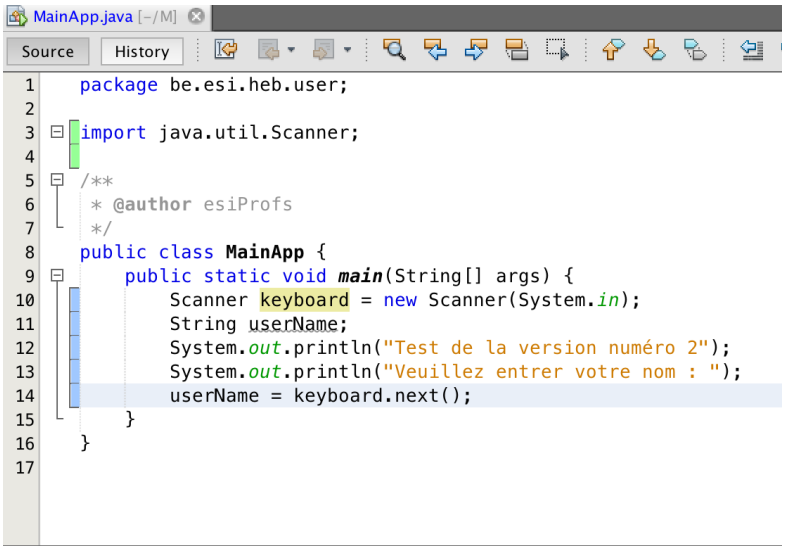
On y trouve également la liste tous les fichiers qui ont été créés, modifiés ou supprimés par votre travail. Comme il s'agit d'un nouveau projet, il s'agit ici des fichiers créés et gérés par NetBeans en plus de la classe `MainApp`.

- 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

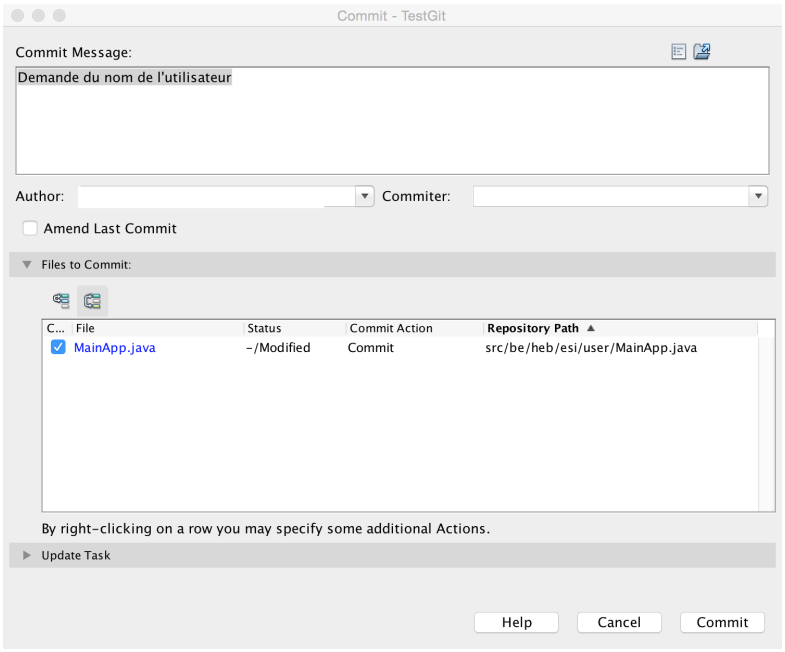
Dans le cadre de ce projet une nouvelle fonctionnalité est demandée, l'utilisateur doit encoder son nom.

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



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`.

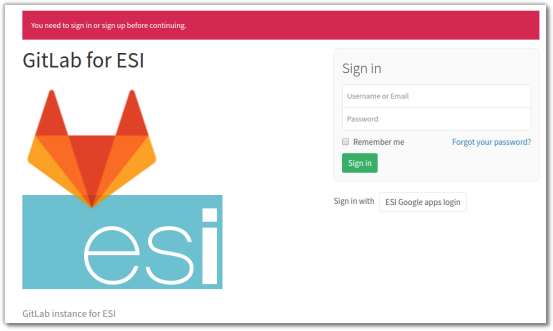
2 Le serveur gitlab

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`.

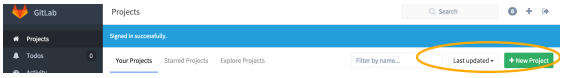
Pour vous faciliter les opérations ultérieures, il est important de définir un mot de passe propre à ce serveur. Pour cela,

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

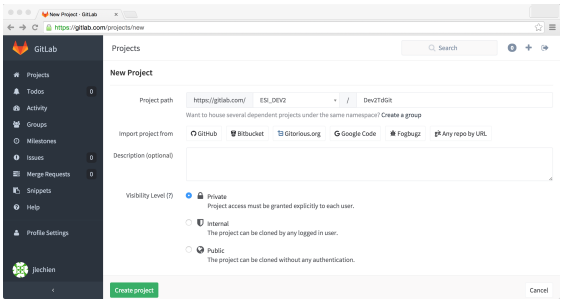
Tutoriel 6 Création d'un dépôt sur le serveur

Créons 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.



Tutoriel 7 Premier push

Il est temps de sauver l'historique de votre projet NetBeans sur le serveur GitLab.

Push

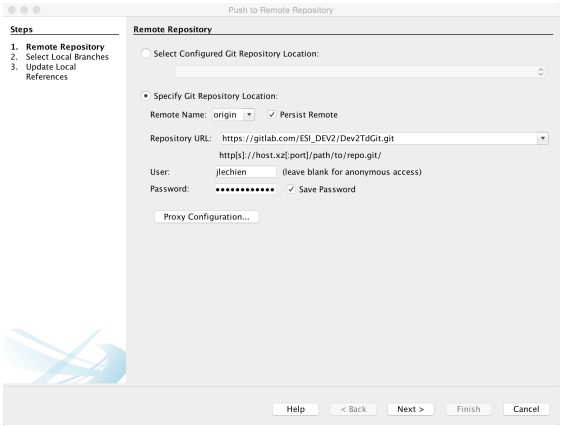
Un **push** consiste à envoyer sur le serveur tous les nouveaux commits effectués localement.

Pour effectuer ce push :

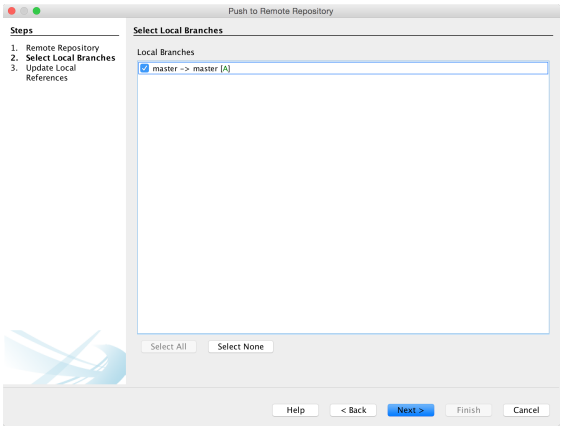
- Sur la page d'accueil de votre dépôt GitLab, récupérez l'url `https` de ce dépôt. Elle sera du type `https://git.esi-bru.be/g12345/Dev1TdGit.git`. Le plus simple est de la copier dans le presse-papier.
- Dans NetBeans :
  - Dans l'onglet `Projects` cliquez droit sur le projet.
  - Choisissez l'option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

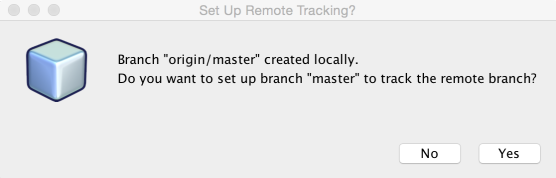
- Dans `Remote Repository` :
  - Copiez l'url *https* de votre projet (celui que vous venez de copier) ;
  - Entrez votre `username`
  - Ainsi que votre `password` (défini un peu plus tôt).
  - Cliquez sur `Next>`



- Dans `Local Branches` :
  - Sélectionnez `master -> master`
  - Cliquez sur `Next>`



Dans `Update Local References` : Il vous suffit de cliquer sur `Finish`
La première fois que vous effectuez un push, un popup apparait. Confirmez sans crainte.



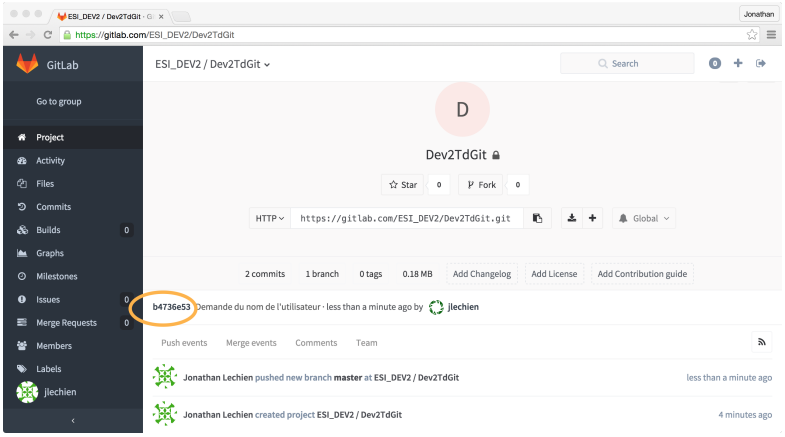
On ne push que ce qui est commité

Attention, lors du *push*, seuls les commits sont envoyés sur le serveur. Si vous avez sauver un fichier mais que vous ne l'avez pas inclus dans un commit, il ne sera pas envoyé sur le serveur !

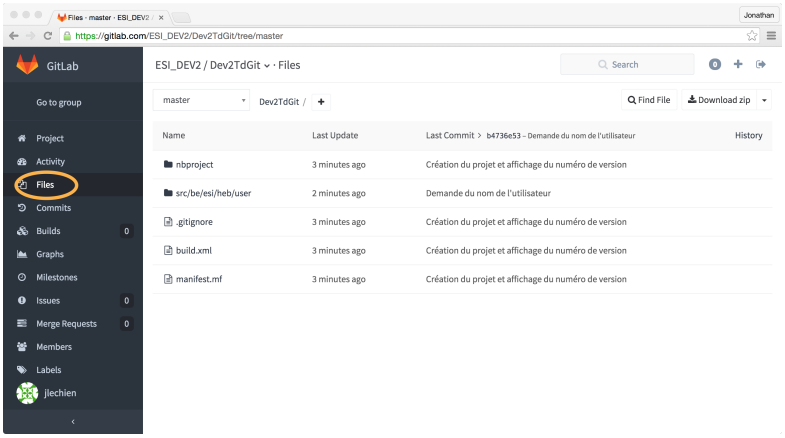
Tutoriel 8 Changement sur le serveur

L'effet du *push* peut être constaté sur le serveur.

Visitez la page principale de votre projet sur le serveur gitlab. Le résultat de votre `push` apparait sur la page d'accueil.



Via le menu `Files`, vous pouvez visualiser les fichiers déposés sur le serveur et leur contenu.



3 Changement de machine

Que se passe-t'il si on se connecte sur une machine où ne se trouve pas encore le projet ? Comment le récupérer pour pouvoir continuer à travailler ? C'est ce que nous allons expérimenter dans cette section.

Pour ce faire, vous devez donc vous déconnecter de cette machine pour vous connecter sur une autre et reprendre le TD à cet endroit.

À tout de suite...

Tutoriel 9

Cloner un projet

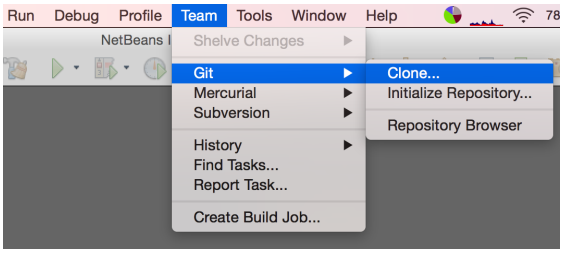
Vous voici sur une nouvelle machine et vous aimeriez récupérer ce qui a été déposé sur le serveur GITLAB.

Clone

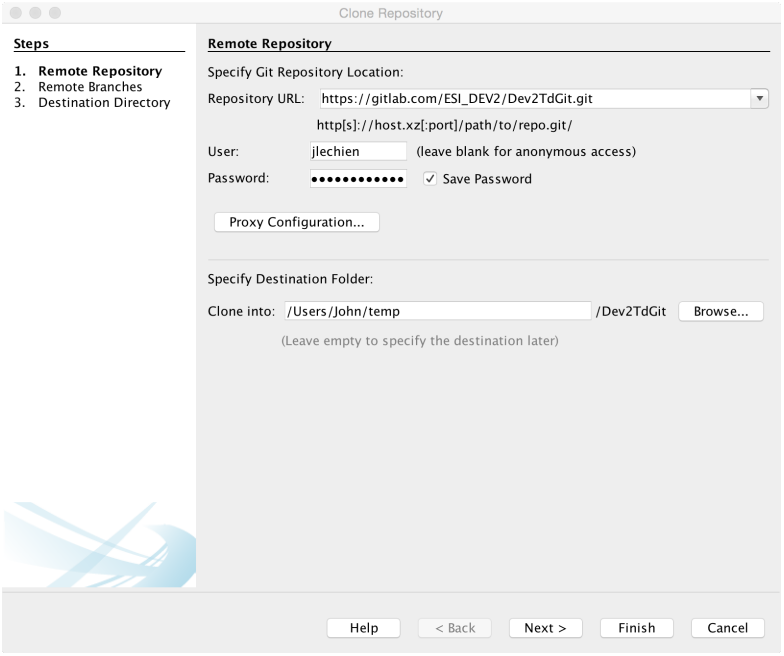
Un **clone** consiste à récupérer de GITLAB [a] le contenu **complet** d'un dépôt (tous les commits). On l'utilise lorsqu'il n'y a encore **rien** sur la machine.

a. ou tout autre serveur du même genre.

- Dans NetBeans, choisissez `Team/Git/Clone...`

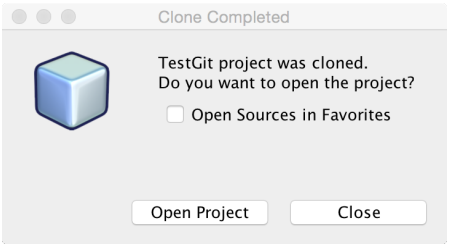


- Vous êtes renvoyé vers l'écran de clonage. Pour terminer l'opération il suffit :
  - Dans `Remote Repository` : d'entrer l'url du dépôt, votre login et mot de passe et de choisir un répertoire pour l'enregistrer ;



  - Dans `Remote Branches` : de sélectionner l'option `master` ;

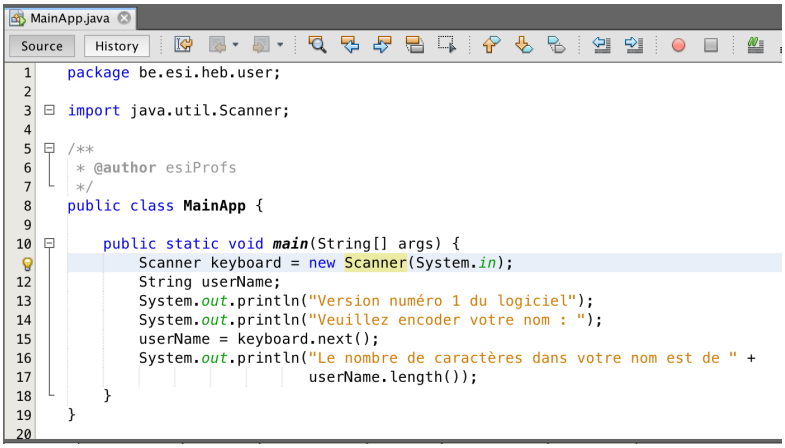
- Dans `Destination Directory` : de cliquer sur `Finish` ;
- Enfin, de confirmer l'ouverture du projet dans NetBeans



Tutoriel 10 **Continuer le projet**

Continuons à développer le projet. Le programme doit désormais afficher le nombre de caractères du nom encodé par l'utilisateur. Vous savez tout ce qu'il faut pour mener à bien cette tâche.

- Modifiez le programme ;



- Effectuez un **commit** ;
- Effectuez un **push** ;
- Allez sur le serveur et vérifiez que vos modifications s'y trouvent.

4 Reprendre un projet

Voyons maintenant ce qui se passe lorsqu'on vous voulez reprendre le travail sur la première machine.

Afin de continuer l'exercice déconnectez-vous de votre machine et revenez à la machine du début du TD.

À tout de suite...

Tutoriel 11 **Pull**

Vous voici de retour sur la machine du début.

- Ouvrez le projet NETBEANS.

Il est important de comprendre que cette version **n'est pas à jour**.

Pull

Un **pull** consiste à mettre à jour le contenu d'un dépôt. On va récupérer du serveur tous les commits manquants.

Pour récupérer les changements depuis le serveur, suivez les étapes ci-dessous.

- Dans l'onglet `Projects` cliquez droit sur le projet.
- Choisissez l'option `Git > Remote > Pull...`.

Par défaut, NETBEANS entre en contact avec le serveur où vous avez effectué vos commits et il récupére les nouveaux commits qu'il y trouve.

- Exécutez le projet, vous constaterez qu'il s'agit bien de la version développée sur l'**autre** machine.

5 En résumé

Voici un résumé des étapes par lesquelles vous devez passer lorsque vous utilisez GIT.

Pour démarrer un projet

1. Créez un projet sur le serveur [1] ;
2. Créez un projet dans Netbeans ;
3. Faites suivre le projet par git ;
4. Faites un premier **commit** ;
5. Faites un **push** pour le lier au projet sur le serveur et envoyer le premier commit.

À chaque fois que vous y travaillez

1. En début de séance vous **devez** faire
   - ▷ un **pull** (si le projet existe déjà en local mais qu'il n'est pas à jour)
   - ▷ ou un **clone** (si le projet n'existe pas du tout sur cette machine).
2. Durant votre travail, dès que vous avez fini une fonctionnalité, vous **devez** faire un **commit**.
3. Après chaque commit ou une fois, en fin de séance, vous **devez** faire un **push**.

Erreur courante

Il est vraiment très important que vous soyez attentifs aux 2 points suivants :

1. Avant de quitter votre poste de travail, faites un **commit** et un **push** de votre projet afin que tout soit sur le serveur.
2. Avant de continuer un projet, faite un **pull** afin de récupérer tout ce qui se trouve sur le serveur.

Si vous ne suivez pas ces consignes, vous aurez des incohérences de dépôt qu'il sera plus difficile de corriger.

1. Cette étape peut aussi se faire juste avant le push.