

Gestion de versions avec IntelliJ, gitlab et git

La gestion de versions (version control ou revision control) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers. Essentiellement utilisée dans le domaine de la création de logiciels, elle concerne surtout la gestion des codes sources et s'applique donc particulièrement bien au projet **Twisk**.

Même si vous êtes seul à développer, la gestion des versions vous permet de revenir très facilement à une version antérieure de votre logiciel et de récupérer votre code et de continuer à développer quelle que soit la machine sur laquelle vous êtes connecté, si toutefois vous utilisez un dépôt distant (une forge). Ce dépôt vous sert également de sauvegarde de votre application.

Ce dépôt distant va également permettre aux enseignants d'avoir accès à votre code source (ce qui dispense ainsi d'un dépôt explicite du code source sur arche) et de suivre votre développement : les mises à jour doivent être fréquentes, régulières, bien commentées et équilibrées entre les membres du binôme.

Dans le cadre du projet de synthèse **Twisk**, nous utilisons un dépôt distant (appelé **forge**) accessible à partir de <https://gitlab.univ-lorraine.fr> ; le logiciel de gestion des différentes versions s'appelle **git**.

Choisissez le projet à déposer. Seul un des étudiants du binôme doit donc faire le dépôt du projet sur la forge, (push - étape 1) l'autre membre du binôme devra effectuer l'étape de récupération (clone - étape 2). Ensuite, chaque membre du binôme pourra déposer et mettre à jour son code à sa guise.

1- Import du projet existant sur la forge, à faire par l'étudiant dont le code a été choisi pour initialiser le dépôt

Sous le projet ouvert sous IntelliJ :

- préparation à la gestion de version avec **git** :
 - *facultatif pour certaines version d'intelliJ : File > Settings > Version Control*
 - * au bout de la ligne **<project>**, indiquez **git** (cliquez sur **<none>** pour pouvoir sélectionner **git**)
 - **VCS > Create Git Repository**
 - * sélectionnez le répertoire de votre projet (nom du projet), cela indique seulement l'emplacement des fichiers locaux créés par **git** pour effectuer la gestion des versions
- import du projet existant sur la forge :
 - **VCS > VCS Operations > Commit**
 - * sélectionnez les fichiers sources à importer sur la forge ; généralement, on sélectionne le répertoire **src** ; **attention** surtout veillez bien à ce que les fichiers de configuration sous le répertoire **.idea** ne soient pas sélectionnés (ces fichiers sont locaux à l'installation de **intelliJ** sur une machine, ils ne doivent pas être recopiés sur d'autres machines) ;
 - * dans la fenêtre **Commit Message**, entrez le message du premier dépôt, par exemple :

Premier dépôt du projet

et cliquez sur le bouton **"Commit and Push"**.

- * cliquez sur **master** -> **Define remote**, ne changez pas le nom **origin**. On vous demande de saisir l'url de votre projet, cette url est à reprendre depuis la forge gitlab :
 - connectez-vous sur le site <https://gitlab.univ-lorraine.fr> et sélectionnez le projet **twisk** déjà présent
 - sélectionnez le bouton **clone** en haut à droite et copiez l'url en https, elle ressemble à :

<https://gitlab.univ-lorraine.fr/coll/sciences-et-technologies/fst/l2-info/projet/login/twisk.git>

- * validez le **"push"**
- * de retour sur la page web de **gitlab** (page à rafraichir), vous pouvez constater que votre code y est déposé

2- Récupération du code depuis la forge, à faire par l'autre étudiant du binôme

- détruisez, en dehors d'intelliJ, votre projet **Twisk**.
- ouvrez **intelliJ** :
 - **"File > New > Project from Version Control > git"**
on vous demande de saisir l'url de votre projet, cette url est à reprendre depuis la forge **gitlab** :
 - ★ connectez-vous sur le site <https://gitlab.univ-lorraine.fr> et sélectionnez votre projet **twisk**
 - ★ sélectionnez le bouton **clone** en haut à droite et copiez l'url en https, elle ressemble à :

https://gitlab.univ-lorraine.fr/coll/sciences-et-technologies/fst/l2-info/projet/login/twisk.git
 - choisissez l'emplacement de votre projet, son nom et cliquez sur le bouton **Clone**

3- Récupération et mises à jour du dépôt distant, à faire par chaque étudiant

Sous le projet **Twisk** ouvert sous **intelliJ** :

- pour valider et envoyer les changements sur le dépôt distant :
 - **VCS > Commit ...**
 - vérifier la liste des fichiers à mettre à jour sur la forge (attention aux fichiers de configuration)
 - écrire un message pertinent
 - **Commit > Commit and Push ... > Push**
- pour récupérer les modifications depuis le dépôt distant :
 - **VCS > Update Project**
- pour connaître les modifications locales :
 - **VCS > Git > Compare with the same Repository version**
- pour écraser la version locale du projet :
 - **VCS > VCS Operations Popup**
 - item **Revert** et sélectionner les fichiers pour lesquels vous voulez oublier les modifications locales
- pour revenir à une version précédente du projet :
 - sélection de la racine du projet dans la liste de gauche
 - **VCS > Git > Show History > Onglet "History : nomProjet"**
 - sélection de la version à restaurer dans la liste des révisions
 - Clic droit : **Reset Current Branch to here**

Documentation

Tout savoir sur la gestion de versions :

http://fr.wikipedia.org/wiki/Gestion_de_versions

Toute la documentation de git avec intelliJ se trouve sur le site :

<https://www.jetbrains.com/help/idea/version-control-integration.html>