

Configuration du poste local

Auteur : Romain NOEL

Date : 05/03/2020



Version : 1


Périmètre de l'opération : Configurer son poste local pour la suite des exercices

Prérequis : N/A

Outil : Git

Durée : 30min

Opérations (ce qu'il faut faire)	Points clés (à quoi il faut faire attention)	Argumentaire (pourquoi c'est important)
1/ Installer un logiciel de gestion de code source en local sur le poste de travail <ul style="list-style-type: none"> Télécharger git for windows: https://gitforwindows.org/ Choisir la pré-configuration d'installation par défaut en cliquant sur next systématiquement. 	<ul style="list-style-type: none"> <input type="checkbox"/> Je vous propose l'utilisation du logiciel GIT Bash en mode ligne de commande (Bash) mais il existe d'autres logiciels possibles comme CYGWIN, ou le mode interface utilisateur. <input type="checkbox"/> La configuration par défaut permet d'avoir GIT BASH (la version en ligne de commandes). <input type="checkbox"/> Si vous le souhaitez, vous pouvez customiser l'installation mais en choisissant au minimum la version en mode ligne de commande (Bash). 	<ul style="list-style-type: none"> ➤ GIT BASH et Cygwin simulent un environnement Unix sous Windows, rendant possible l'exécution de ces logiciels après une simple compilation. ➤ Ils permettent de faire des commandes "git" pour pousser/télécharger le code entre le poste de travail et le serveur Gitlab central. ➤ Ils permettent le transfert des fichiers vers gitlab central avec le protocole GIT.
2/ Créer un répertoire en local <ul style="list-style-type: none"> Ouvrir le logiciel GIT Bash  Vérifier où on se trouve dans l'arborescence : <code>pwd</code> Se positionner dans le répertoire dans lequel on veut créer le répertoire du projet : <code>cd /c/<workspace></code> 	<ul style="list-style-type: none"> <input type="checkbox"/> Il est recommandé de le créer sur le disque dur pour éliminer les latences réseaux. <input type="checkbox"/> Cette pratique doit être couplée à des commits réguliers et des pushes sur le dépôt central. Rien ne doit rester sur le disque dur. 	<ul style="list-style-type: none"> ➤ Créer l'espace de travail dans lequel sera localisé le code source en local.
3/ Configurer les paramètres <ul style="list-style-type: none"> Ouvrir le logiciel GIT Bash  <code>git config --global user.name "<prénom nom>"</code> <code>git config --global user.email "<adresse mail>"</code> <code>git config --global http.sslVerify false</code> Vérification des données déclarées dans le fichier config <code>git config --global -l</code> 	<ul style="list-style-type: none"> <input type="checkbox"/> A faire une fois à l'installation de l'outil 	<ul style="list-style-type: none"> ➤ Permet d'avoir son nom plutôt qu'un identifiant associé à son activité dans les différents outils. ➤ La mise à false du paramètre http.sslVerify qui désactive la vérification du certificat ssl et élimine les logs sur le sujet .

<p>4/ Se connecter au référentiel de code source GitHub Central</p> <ul style="list-style-type: none"> • Se connecter au portail GitHub https://github.com/ • Cliquer sur sign up et se créer un compte (dans le cas où vous avez déjà un compte, connectez-vous simplement en cliquant sur sign in) • Cliquer sur « new » dans la colonne de gauche • Donner un nom à votre projet puis cliquer sur « create repository » • Copier l'URL de votre projet 	<ul style="list-style-type: none"> ❑ L'onglet Standard est réservé aux administrateurs Git. 	<ul style="list-style-type: none"> ➤ La page d'accueil Gitlab contient la liste des projets dont vous êtes membre de l'équipe. ➤ L'url est nécessaire ultérieurement pour la configuration du gitlab local.
<p>5/ Ajouter votre binôme en tant que développeur</p> <ul style="list-style-type: none"> • Entrez dans l'onglet « settings » • Cliquez sur « manage access » dans la colonne de gauche • Cliquer sur le bouton « Invite a collaborator », entrez le pseudonyme de votre binôme puis validez en cliquant sur « add ». • Demander (gentiment) à son binôme d'accepter la demande par mail. 	<ul style="list-style-type: none"> ❑ L'onglet « Settings » permet de gérer tout ce qui est externe au code (droits d'accès, suppression etc...) 	<ul style="list-style-type: none"> ➤ Par défaut, personne ne peut modifier votre projet (même s'il est public). Il faut ajouter les collaborateurs manuellement. ➤ Certains exercices vous demanderont de travailler en binôme sur un même dépôt. Il est donc nécessaire d'ajouter les différents contributeurs au projet.
<p>6/ Créer le repository local et le lien vers le repository central GitHub du projet</p> <ul style="list-style-type: none"> • Ouvrir le logiciel GIT Bash  • Créer le clone sur le poste de travail <p>git clone <copie de l'url></p> <ul style="list-style-type: none"> • Vérifier la création du répertoire au nom du projet <p>ls -l</p> <ul style="list-style-type: none"> • Vérifier la création du fichier .git: <p>cd <nom_du_projet></p> <p>ls -al</p>	<ul style="list-style-type: none"> ❑ git-clone remplace les commandes git init et git remote add qui sont inclus dans le git clone. ❑ Le répertoire au nom du projet est créé. ❑ Le paramètre <copie de l'url> est l'url copié en action 4/ ❑ Le répertoire .git a été créé 	<ul style="list-style-type: none"> ➤ Crée l'instance de gitlocal sur le poste de travail et le lien permettant le transfert entre le repository local et le repository central GitHub du projet. ➤ La création du dépôt local est nécessaire car Git est un outil décentralisé. Il travaille avec un dépôt distant et un local sur le poste de travail permettant d'être indépendant. ➤ L'espace du gitlocal ainsi créé sur le disque dur du poste de travail est un clone de ce qui se trouve sur le gitlab source central.