**Mémo Intellij**

**Plugin si projet Eclipse**

**Adapter for Eclipse Code Formatter**

Permet d'utiliser le formateur de code Java d'Eclipse directement à partir d'IntelliJ.

Nécessaire, voir indispensable, si le projet utilise des configurations de lancement (.launch).

Pas utilisé pour les projets TRS\_commun et TRS\_preva.

Pas testé personnellement.

<https://github.com/krasa/EclipseCodeFormatter#instructions>

**Plugins intéressants**

RainbowCSV facilite la lecture de fichiers .csv

Rainbow Brackets facilite la visibilité de l’indentation

Maven Helper

Maven Dependency Helper

Docker

PDF permet de lire un .PDF

Lombok permet de générer les constructeurs, getter, setter

Ideolog permet de mettre en couleur les états debug/info/warn/error/fatal

**Importer les projets trs\_preva et trs\_commun**

Créer un projet « preva » en important une source existante

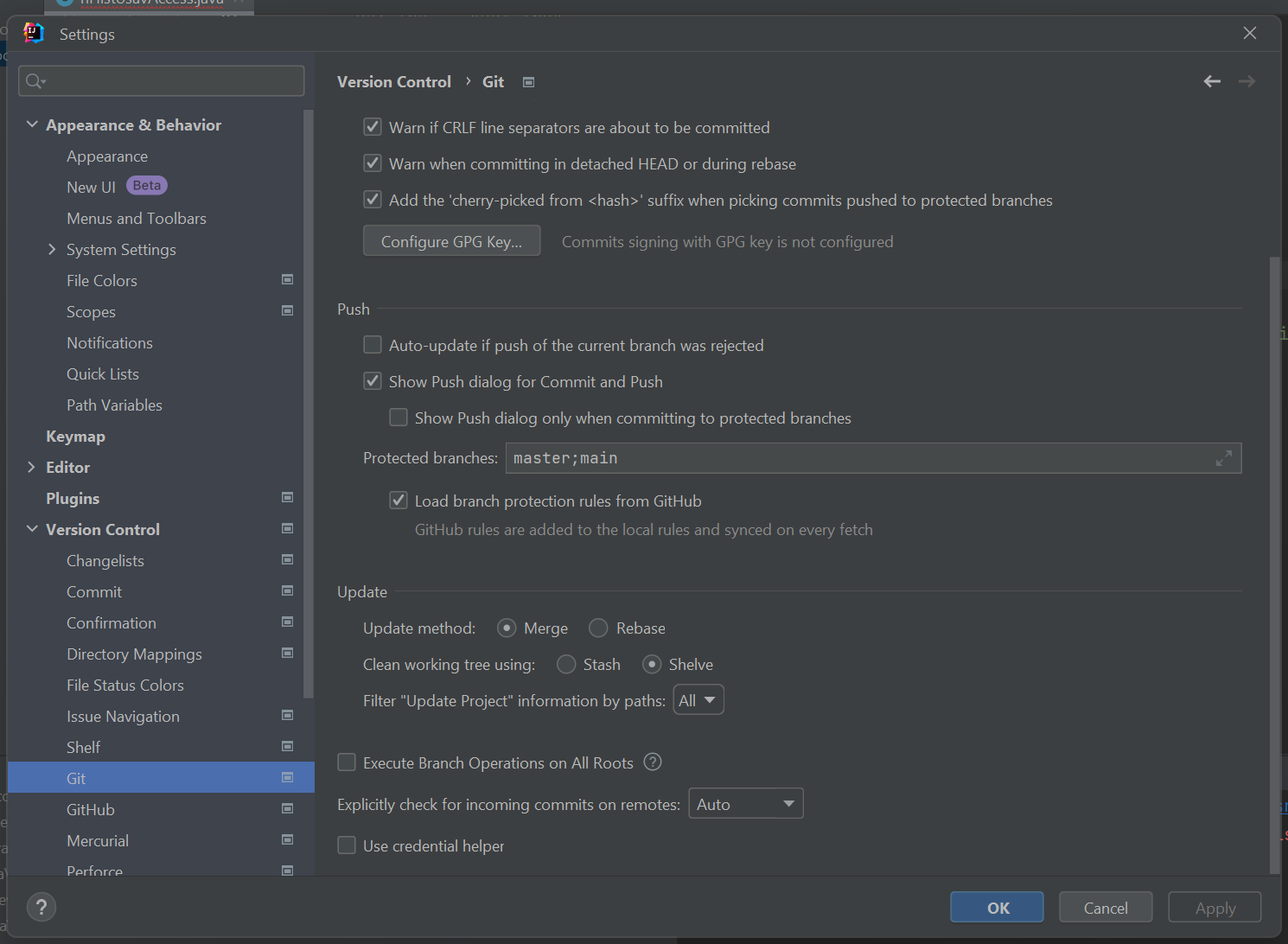
* Sélectionner le dossier souhaité 🡪 ok
* Import project from external model : Eclipse 🡪 next
* Cocher Link created Intellij to eclipse 🡪 next
* Sélectionner le projet 🡪 next 🡪 create

File / Project structure / modules 🡪 + / importer un module

* Sélectionner le dossier souhaité 🡪 ok
* Import project from external model : Eclipse 🡪 next
* Cocher Link created Intellij to eclipse 🡪 next
* Sélectionner le projet 🡪 next 🡪 create

*Voir la suite page suivante*

Configurer IntelliJ IDEA pour ne pas effectuer toutes les opérations de branche (telles que l'extraction, la fusion, la suppression, etc.) simultanément sur toutes les racines comme s'il s'agissait d'un référentiel unique.



**Ne pas cocher**

**Raccourcis clefs**

CTRL + B ou CTRL clic sur usage : affiche la déclaration

CTRL + <espace> : affiche des propositions de nommage

Shift 2 fois : ouverture fenêtre de recherche dans l’éditeur et/ou le projet

CTRL + Shift + F : ouverture fenêtre de recherche d’un texte dans un fichier

Alt + Insert dans pom.xml (maven) : sous-menu très intéressant

**Divers**

Settings : l’icône  indique que le paramètres est lié au projet, sinon quelque soit le projet.

**Création .jar**

Initialiser la construction :

File / Project Structure... / Artifacts

Cliquer sur le "+"

Sélectionner "JAR -> From modules with dependencies..."

Sélectionner la "Main Class" 🡪 ok 🡪 "Apply" 🡪 "OK"

Lancer la construction du .jar :

Build / Build Artifacts...

Sélectionnez "Build";

le fichier "jar" est généré dans le répertoire "out/artifacts/…