



Eclipse-Git-2 : Initialisation (par le chef de projet) d'un projet partagé avec Eclipse / Egit / GitHub

- 1- Répartition des rôles
- 2- Configuration initiale d'Eclipse
- 3- Partage du projet et création d'un dépôt local (local repository)
- 4- Indication des fichiers à ignorer par Git
- 5- Premier Commit sur le dépôt local
- 6- Création d'un jeton Token
- 7- Recopie du dépôt local sur le dépôt distant (remote repository)
- 8- On fait le point ?

Pré-requis : vous êtes le chef de projet et on suppose :

- que vous avez créé un projet nommé **tracegps** sous Eclipse, dans votre workspace local.
- que vous avez créé un compte **delasalle-sio-xxxx-x** (où xxxx-x est votre nom) sur la forge logicielle **GitHub**.
- que vous avez créé un dépôt GitHub nommé **tracegps** (il est important de donner le même nom au projet Eclipse et au dépôt GitHub).
- que vous avez invité vos collaborateurs GitHub à participer au projet.
- que vos collaborateurs ont accepté cette invitation.

Dans ce document, le chef de projet s'appelle **yves-zenels**, mais vous remplacerez bien sûr ce nom par votre vrai compte.

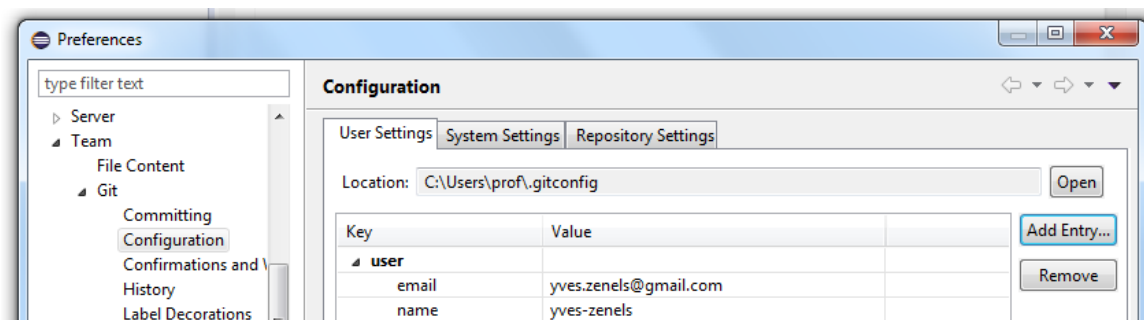
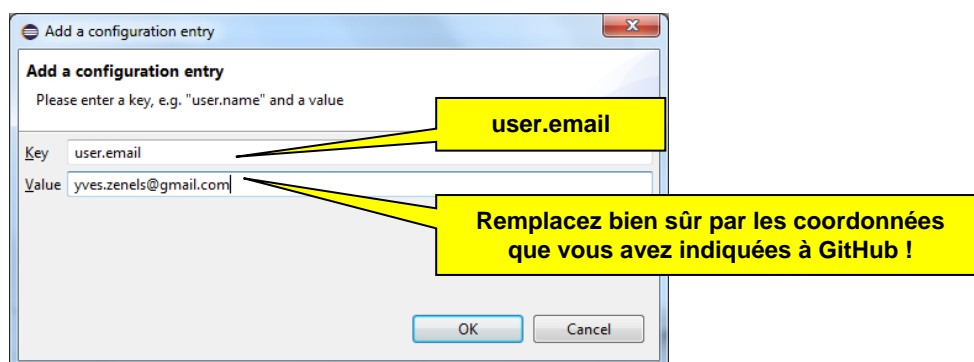
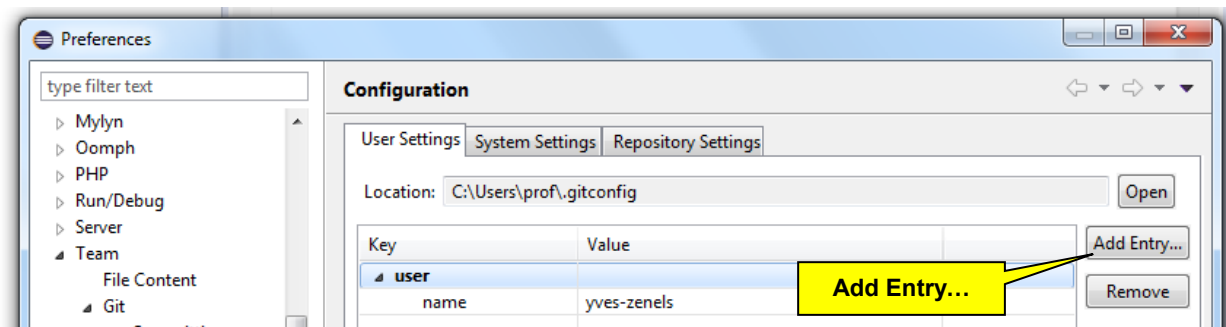
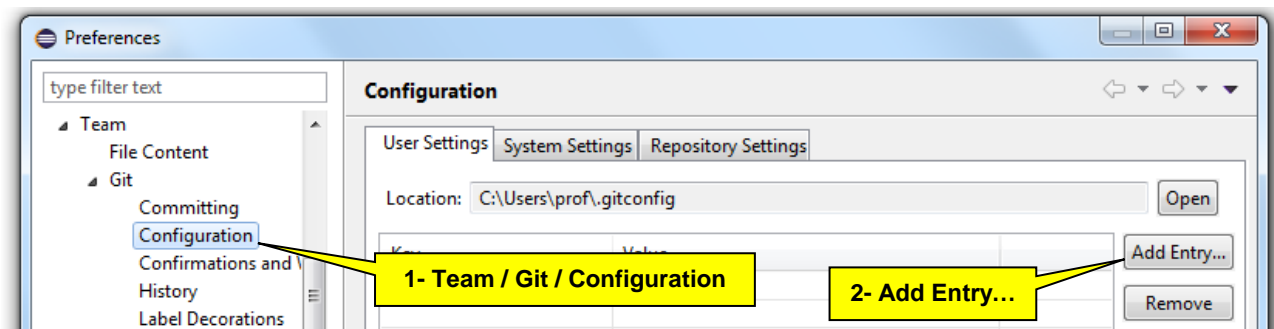
1- Répartition des rôles

Ce document s'adresse au chef de projet mais les autres membres ont bien sûr intérêt à suivre la manipulation avec lui car ils seront sûrement un jour dans la situation du chef de projet.

2- Configuration initiale d'Eclipse

- Démarrez Eclipse et si besoin, indiquez l'emplacement du workspace (l'espace de travail dans lequel Eclipse stocke les projets) avec la commande **File / Switch Workspace** (Eclipse redémarre automatiquement).
- Dès que Eclipse est redémarré, choisissez la **perspective PHP** (et fermez la **perspective Java** avec un clic droit et la commande **Close**)
- Réglez les préférences avec la commande **Window / Preferences** :
 - Choisissez un encodage **UTF-8** avec l'onglet **General / Workspace** et réglez également **New text file line delimiter** à la valeur **Unix**
 - Désactivez (si besoin) la correction d'orthographe avec l'onglet **General / Editors / Text Editors / Spelling**

Enregistrez-vous comme utilisateur de Git en local avec la commande **Window / Preferences** et l'onglet **Team / Git / Configuration** :

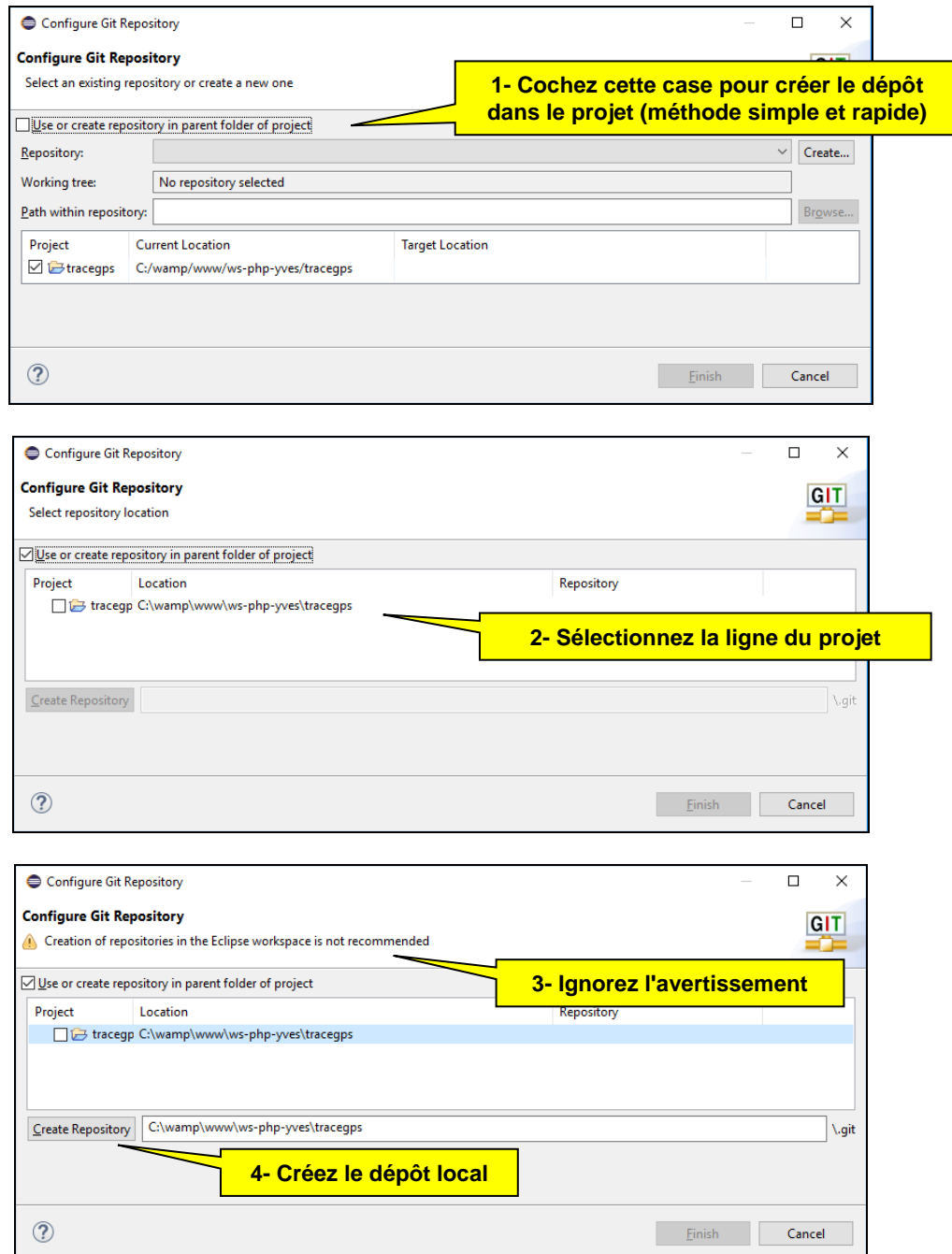


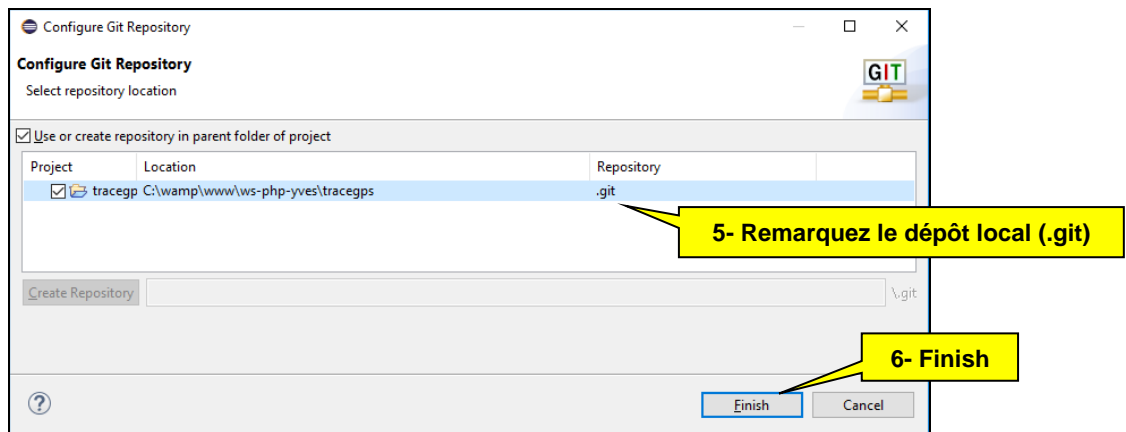
3- Partage du projet et création d'un dépôt local (local repository)

Vous avez déjà créé le projet **tracegps** sous Eclipse, dans votre workspace PHP.

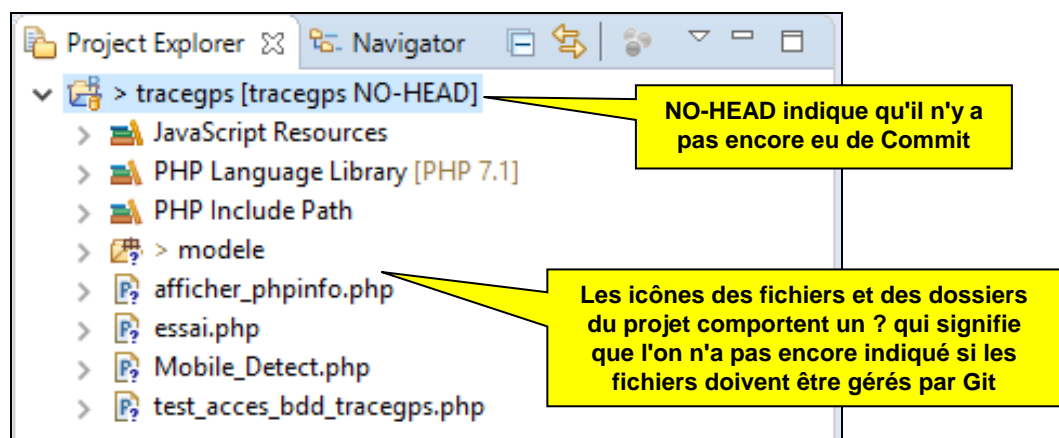
Il faut maintenant créer un **dépôt local** pour le projet, dans lequel Git stockera toutes les informations qui concerneront le suivi de l'état des fichiers du projet.

Dans l'explorateur de projets, faites un clic droit sur le projet et sélectionnez **Team / Share Project...** :

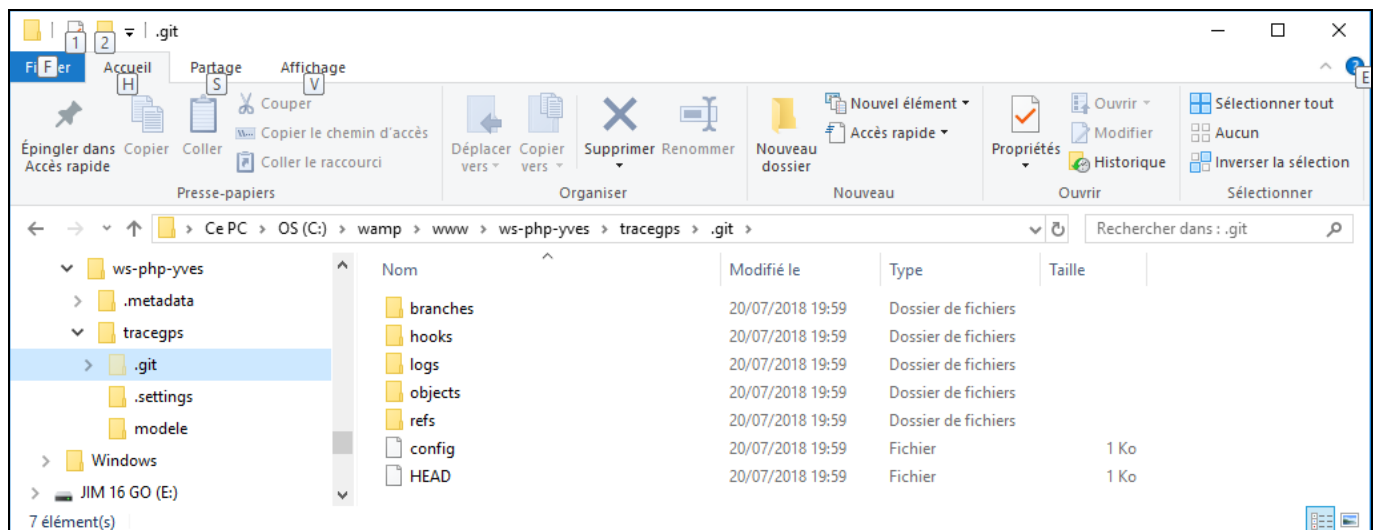




Après quelques secondes (ou après un **F5**), l'affichage du projet est modifié dans le **Project Explorer** :



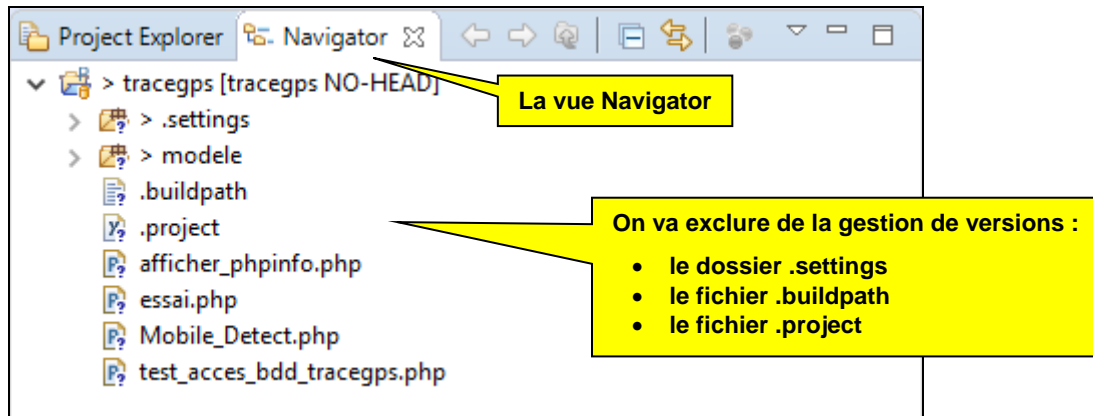
En affichant les dossiers cachés dans l'Explorateur Windows, on peut vérifier la présence du dépôt local (le dossier **.git**) dans l'arborescence du projet (et donc dans le workspace d'Eclipse) :



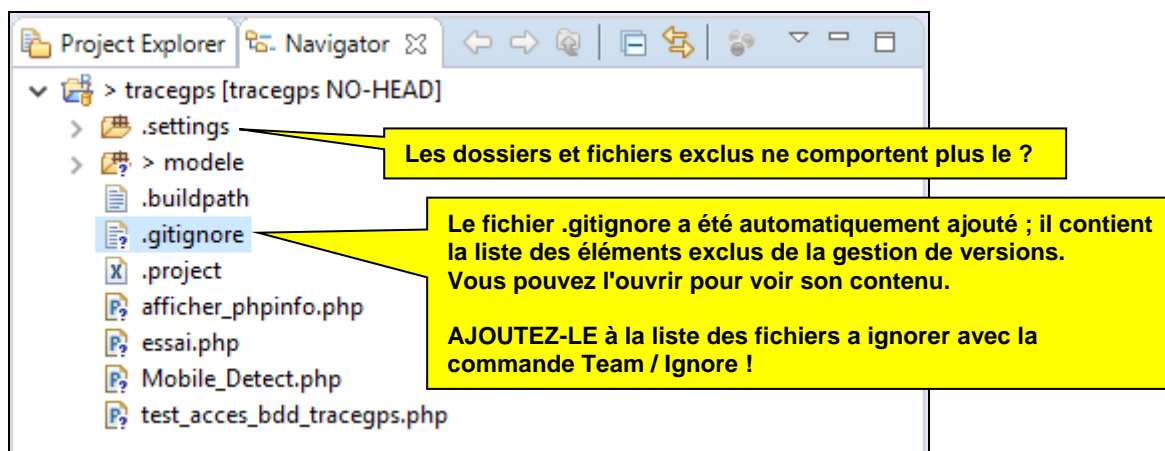
4- Indication des fichiers à ignorer par Git

Certains fichiers ou dossiers propres à chaque développeur ne nécessitent pas de mise en commun et n'ont donc rien à faire dans le repository des sources ; ils peuvent donc être exclus de la gestion de versions (en général, on ne partage que les fichiers de code source).

Pour exclure des fichiers ou des dossiers de la gestion de versions, il suffit d'afficher la vue **Navigator** avec la commande **Window / Show View / Other... / General / Navigator** qui permet de voir tous les fichiers du projet.



Pour chaque fichier ou dossier à exclure, on fait un clic droit et on sélectionne la commande **Team / Ignore** (le point d'interrogation disparaît alors, et GIT sait qu'il doit les ignorer).



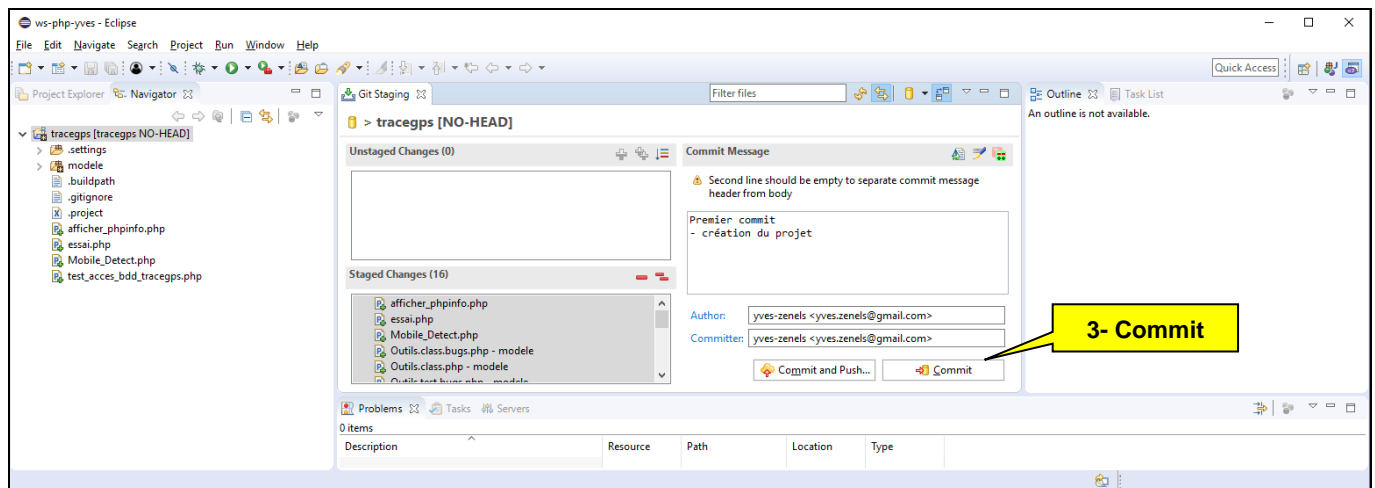
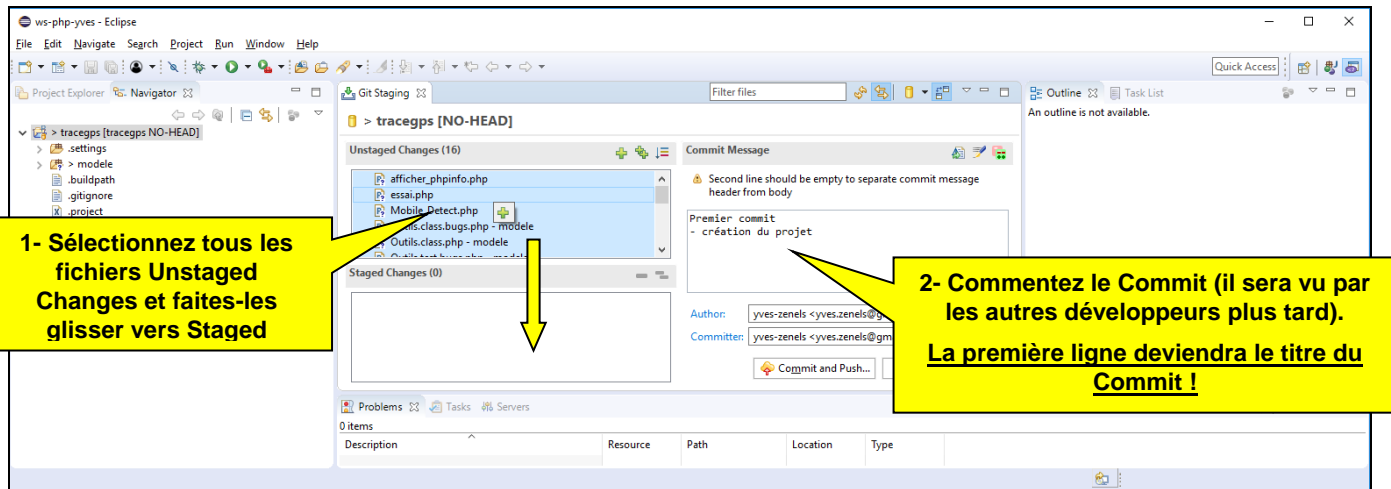
Contenu du fichier **.gitignore** :

```
/.settings/  
/.buildpath  
/.project  
/.gitignore
```

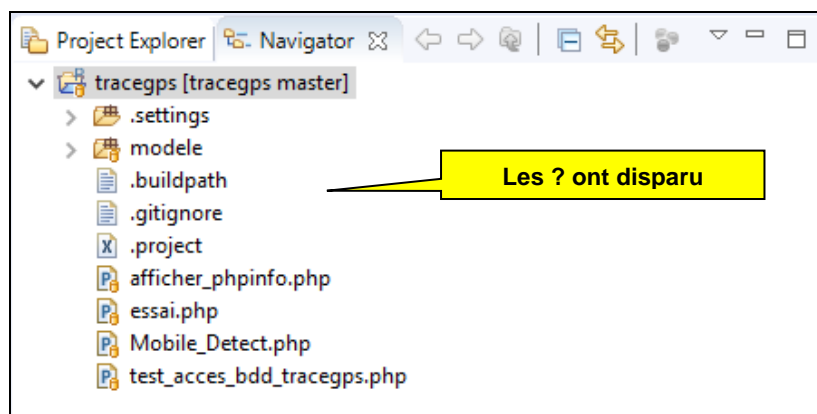
5- Premier Commit sur le dépôt local

Effectuez un premier **Commit** du projet dans le dépôt local avec la commande **Team / Commit...**

Commentez-le **Commit** et sélectionnez les fichiers à "commiter" :



Constatez que les icônes des fichiers et dossiers sont modifiées :



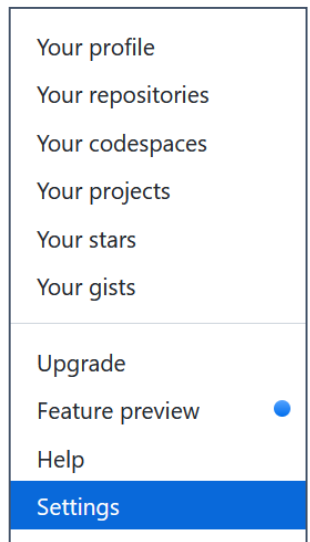
Le projet est maintenant présent sur le dépôt local.

La prochaine étape consistera à recopier le dépôt local vers le dépôt distant (sur GitHub) afin de pouvoir le partager avec d'autres développeurs.

6- Création d'un jeton Token gitHub

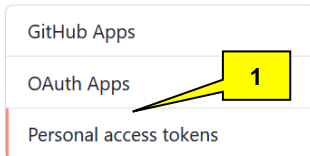
Dans le coin supérieur droit de n'importe quelle page, cliquez sur votre photo de profil, puis sur **Settings**.

Dans la barre latérale gauche, cliquez sur **Developer settings**.



1- Dans la barre latérale gauche, cliquez sur **Personal access tokens**.

2- Cliquez **Generate new token**.



Personal access tokens

2

Generate new token

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

3- Donnez à votre jeton un nom descriptif.

A screenshot of the GitHub 'Generate new token' form. The form has a text input for the token name, a dropdown for expiration, a warning message, a 'Select scopes' section, and a 'Generate token' button. Yellow callout boxes provide instructions: '3- Donnez à votre jeton un nom descriptif.' points to the token name input; '4- Pour donner une expiration à votre jeton, sélectionnez le menu déroulant Expiration, puis cliquez sur « No expiration »' points to the expiration dropdown; '5- Vous pouvez ensuite sélectionner les champs d'application, ou permissions, que vous souhaitez accorder à ce jeton. Pour utiliser votre jeton pour accéder aux dépôts à partir de la ligne de commande, sélectionnez « repo »' points to the 'repo' scope checkbox.

monJeton

What's this token for?

Expiration *

No expiration ▾ The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more](#)

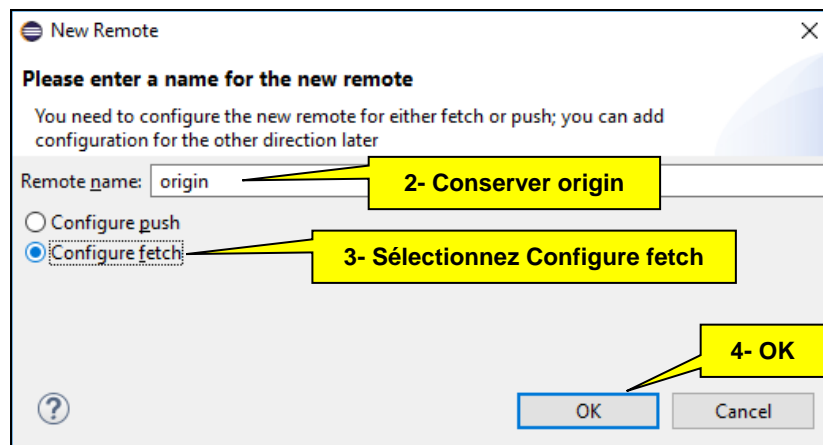
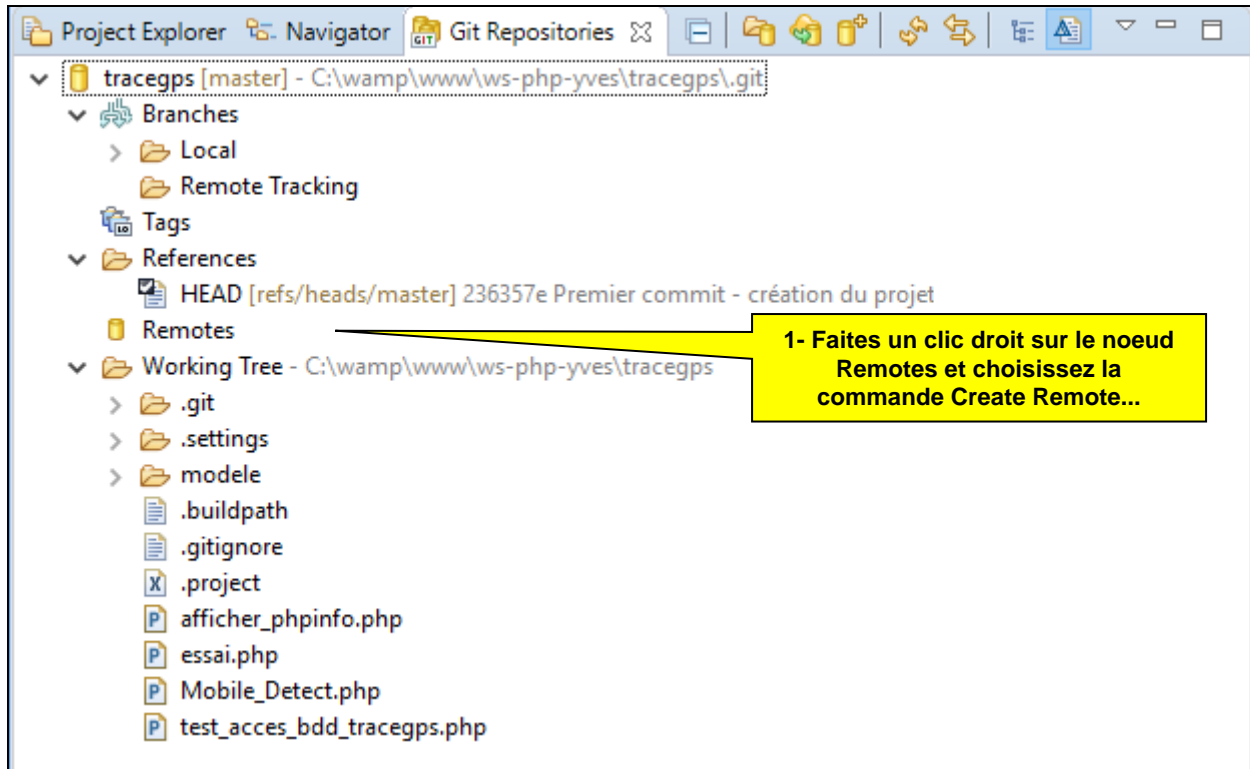
<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

Cliquez ensuite sur **Generate token** et copiez le jeton dans un fichier texte. Vous en aurez besoin pour vous authentifier par la suite.

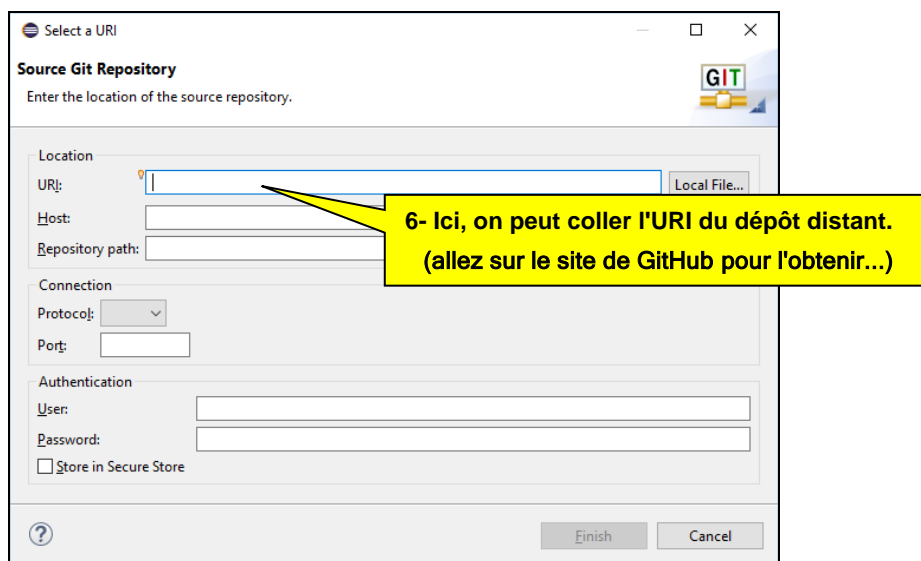
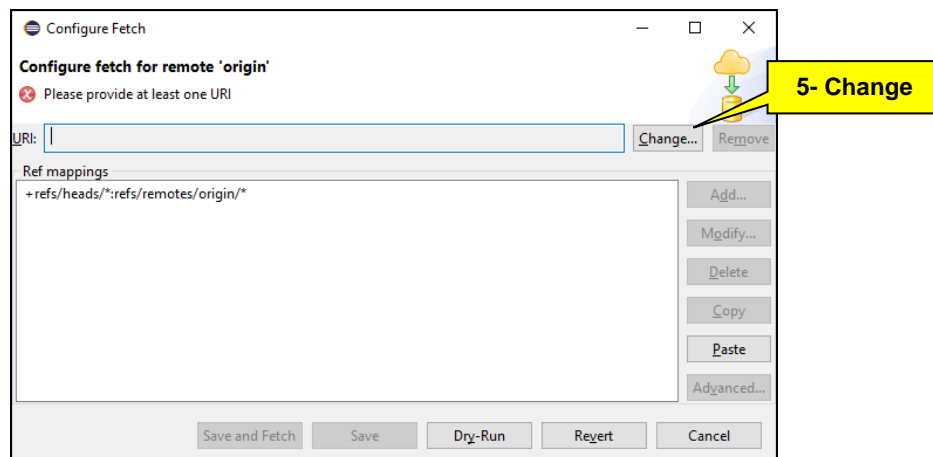
7- Recopie du dépôt local sur le dépôt distant (remote repository)

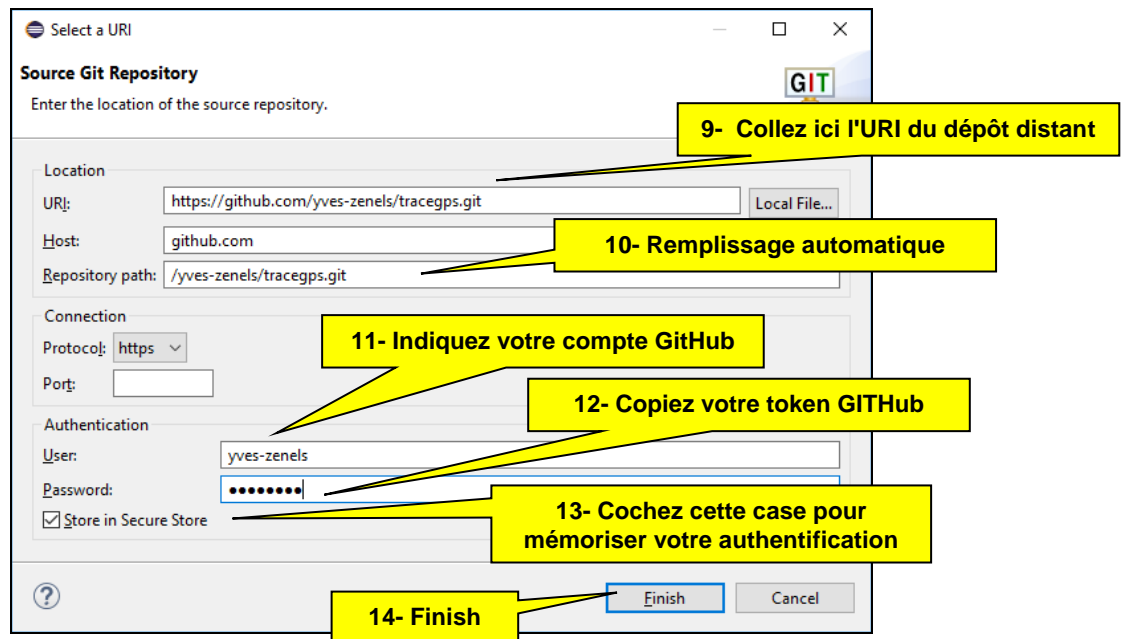
Nous allons maintenant travailler avec la vue **Git Repositories** qui contient la liste des dépôts.

Affichez-la avec la commande **Window / Show View / Other... / Git / Git Repositories** et placez cette vue à un endroit qui vous convient ; déroulez le dépôt **tracegps** pour étudier sa structure :

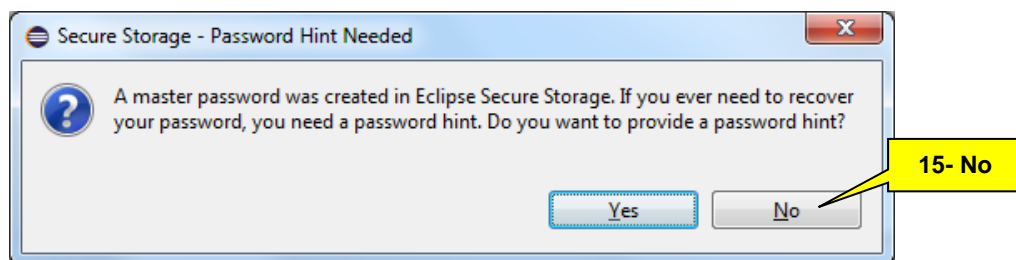


La boîte de dialogue suivante demande l'URI du dépôt distant :

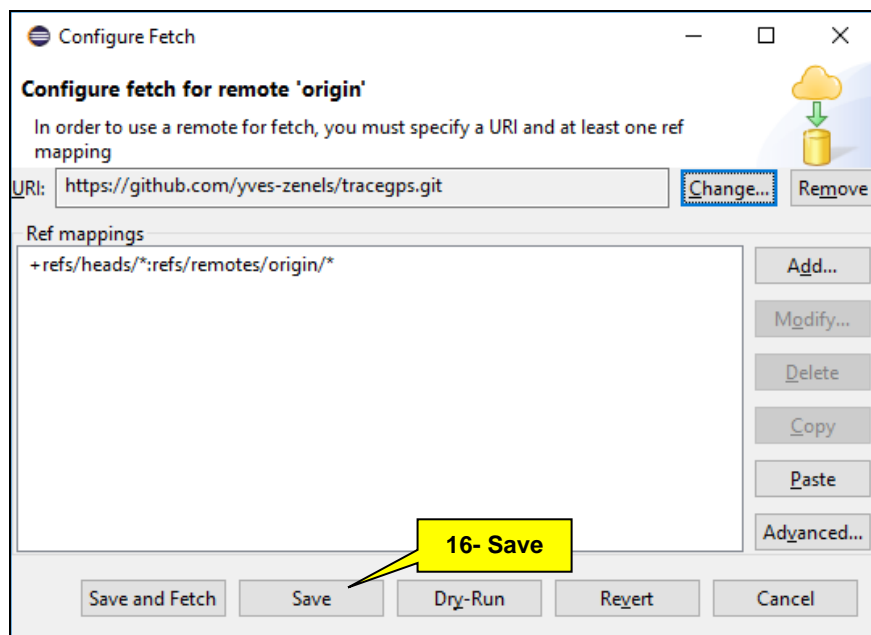




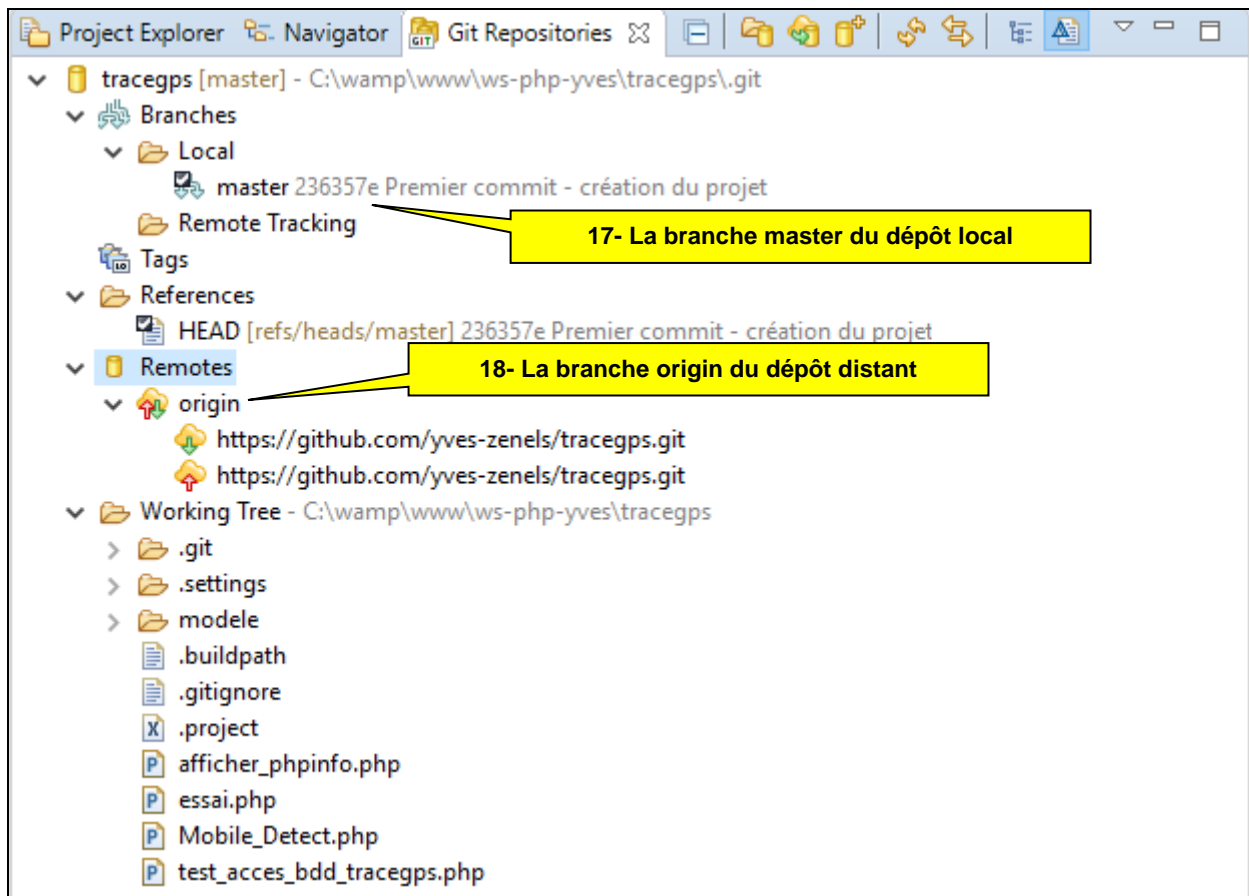
Cette boîte de dialogue peut apparaître :



La boîte de dialogue suivante fait le lien entre les branches du dépôt distant et les branches du dépôt local :



La vue **Git Repositories** a évolué :



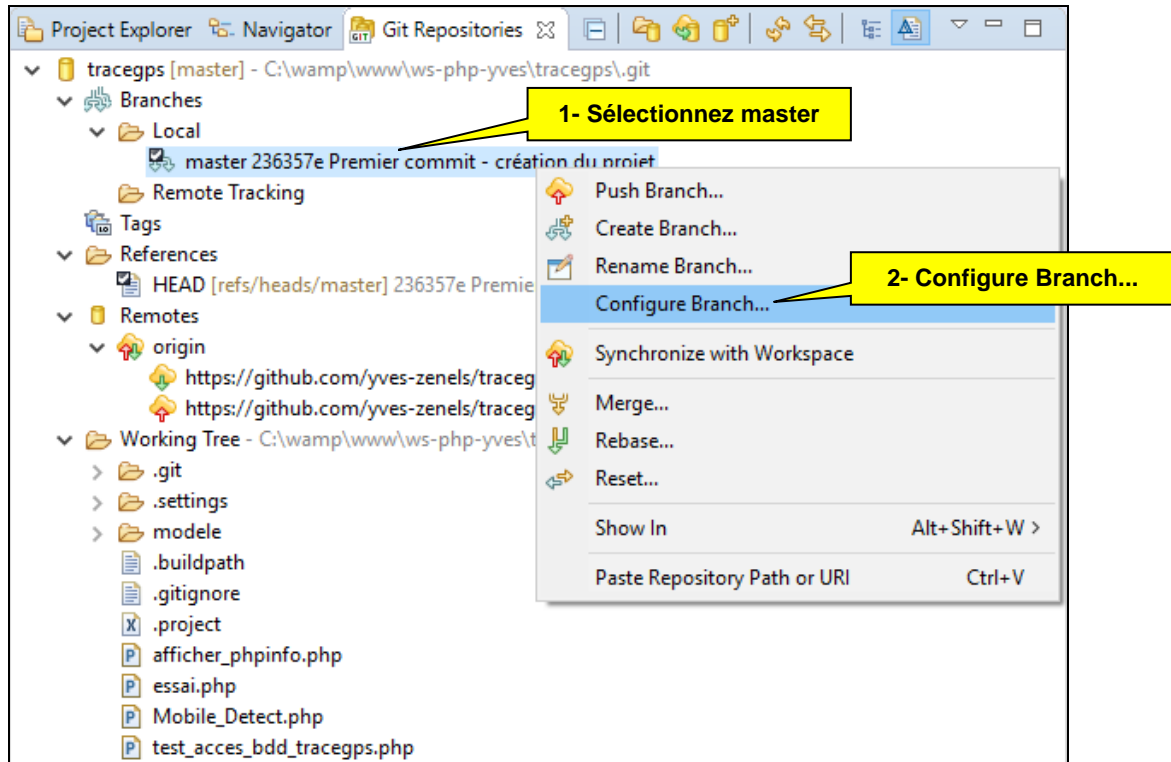
Arrivé à ce point :

- nous disposons d'un dépôt local pour notre projet
- nous avons indiqué à EGit l'adresse de notre dépôt distant pour ce projet

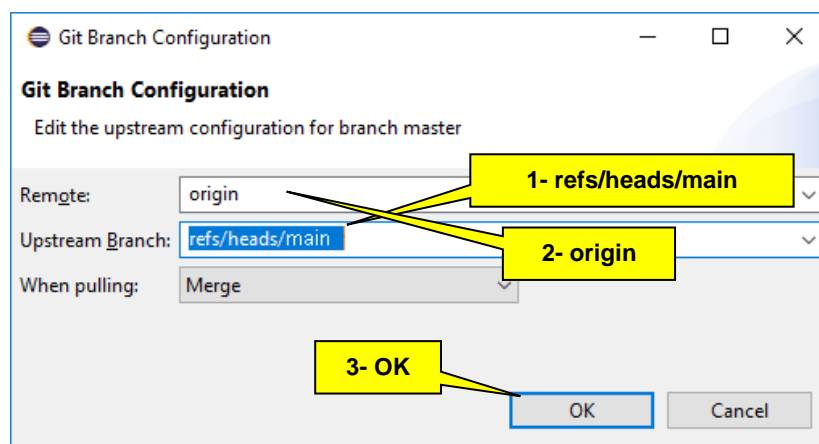
Toutefois, on ne peut pas encore faire de **push** ni de **pull**, car on n'a pas encore interagi avec le dépôt distant ; du coup EGit ne connaît pas son contenu (branches, commits...).

Si on essaye de faire l'un ou l'autre, on risque de se retrouver avec une erreur.

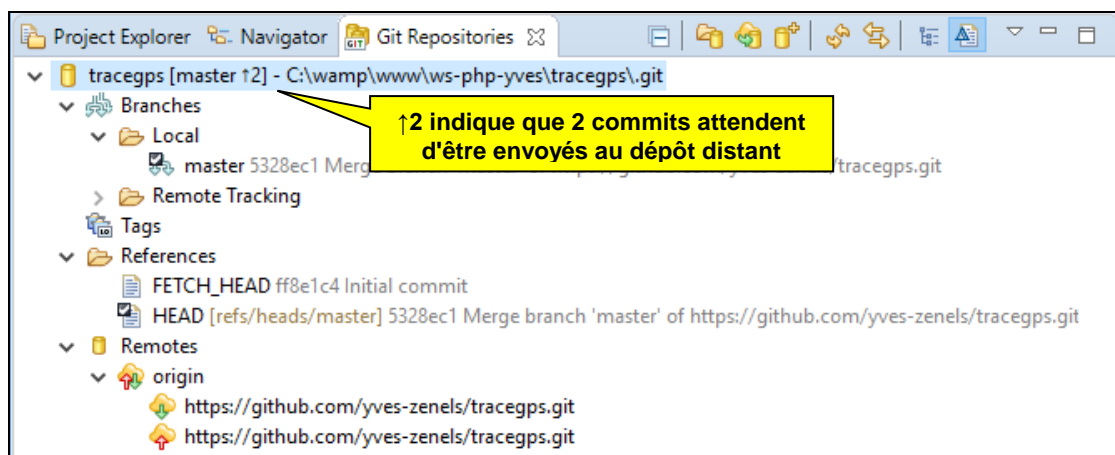
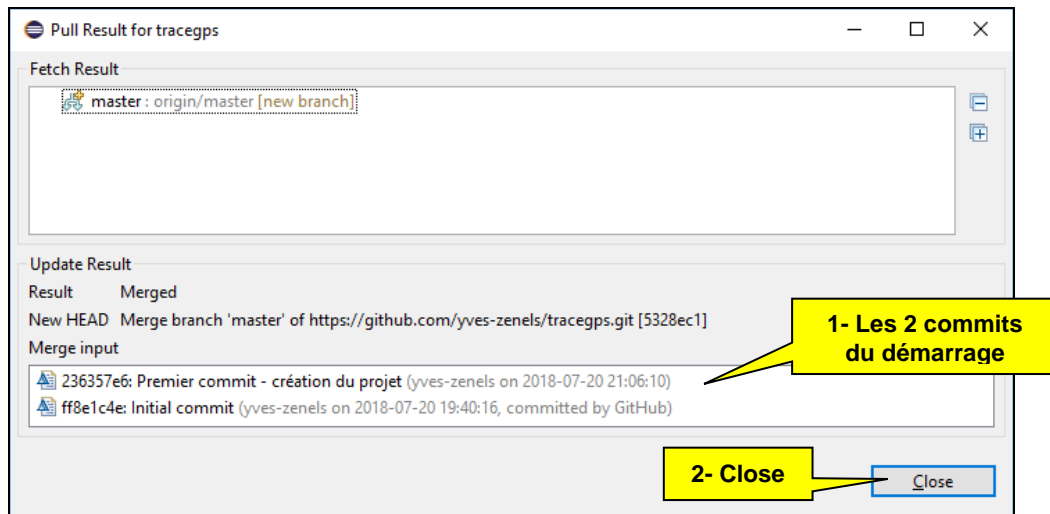
Toujours dans la vue **Git Repositories**, dans votre dépôt, sélectionnez l'élément **master** de l'élément **Branches / Local**, faites un clic droit, et sélectionnez la commande **Configure Branch...** :



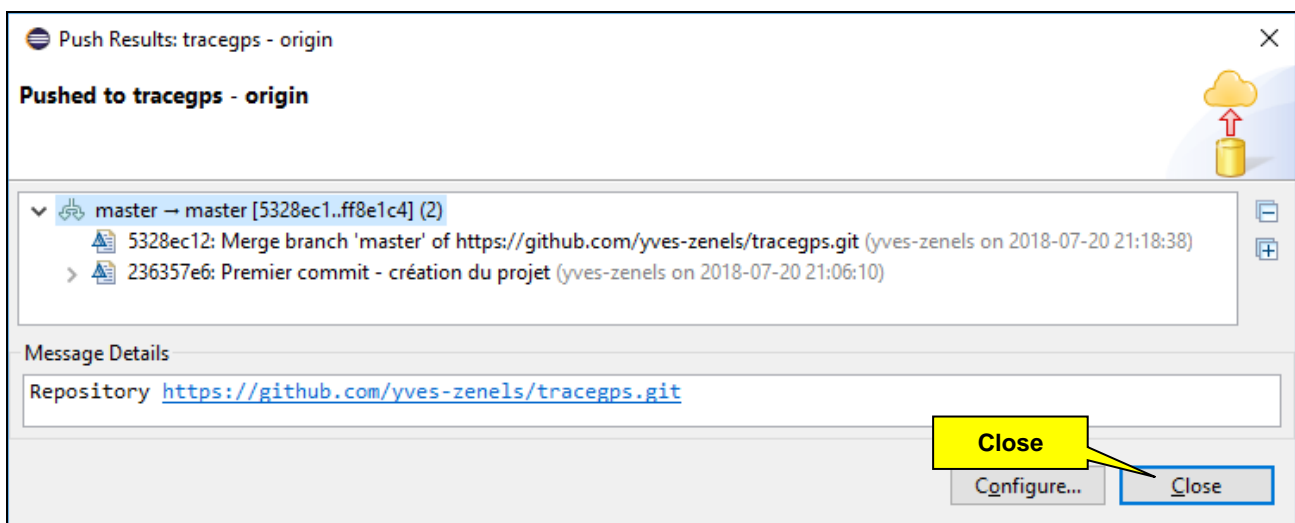
Dans la boîte de dialogue suivante, **réglez le deuxième paramètre AVANT le premier** (c'est plus pratique) en utilisant les 2 listes déroulantes :



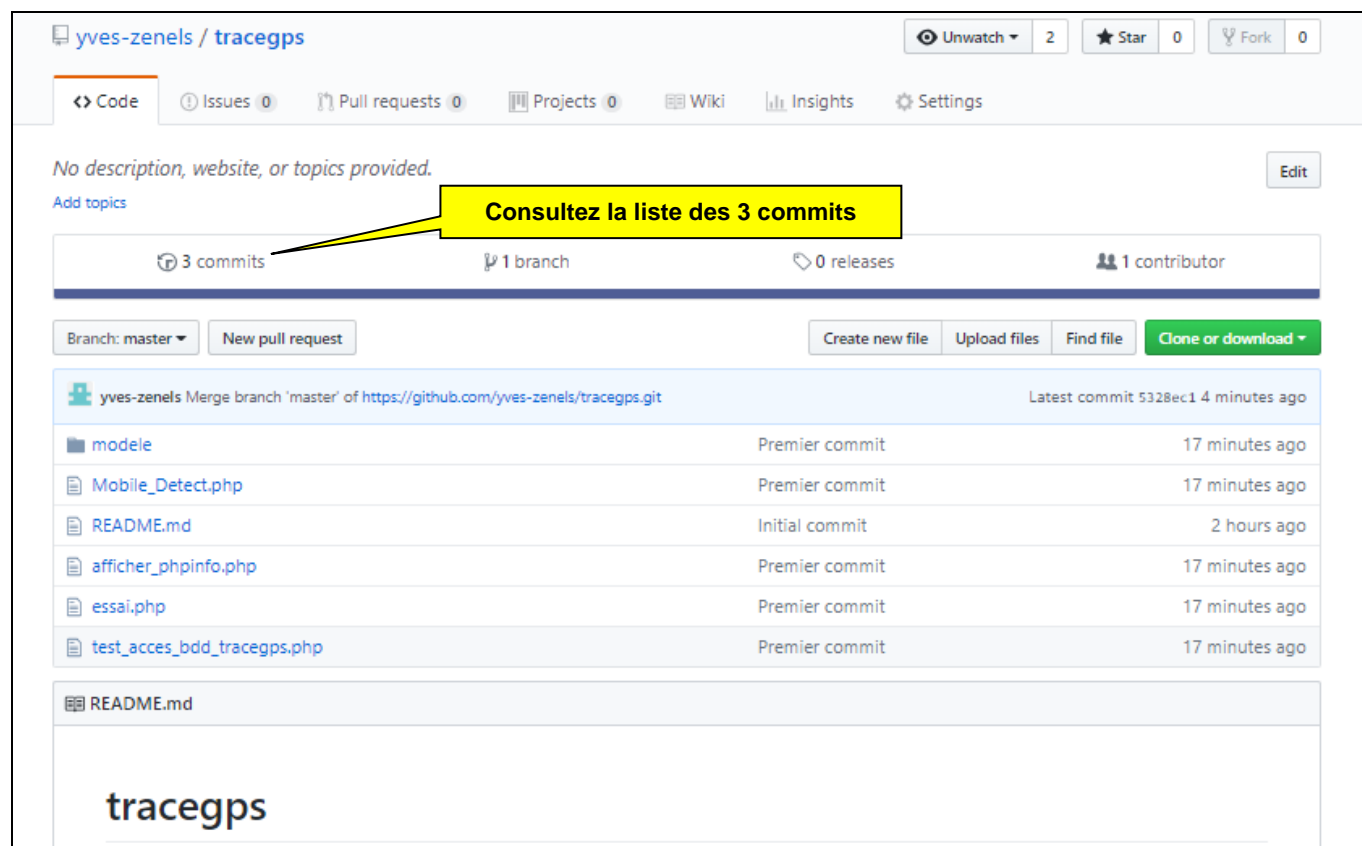
Ensuite, faites un clic droit sur votre projet et sélectionnez la commande **Pull** si vous êtes toujours dans la vue **Git Repositories**, ou la commande **Team / Pull** si vous êtes dans les vues **Navigator** ou **Project Explorer** :



Pour envoyer les 2 commits au dépôt distant, faites un clic droit sur votre projet et sélectionnez la commande **Push to Upstream...** si vous êtes toujours dans la vue **Git Repositories**, ou la commande **Team / Push to Upstream...** si vous êtes dans les vues **Navigator** ou **Project Explorer** :



Il n'y a plus qu'à actualiser la page **GitHub** du dépôt distant **tracegps** pour constater l'arrivée des dossiers et fichiers du premier **Commit** :



Repository: yves-zenels / tracegps

Unwatch 2 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. [Add topics](#) [Edit](#)

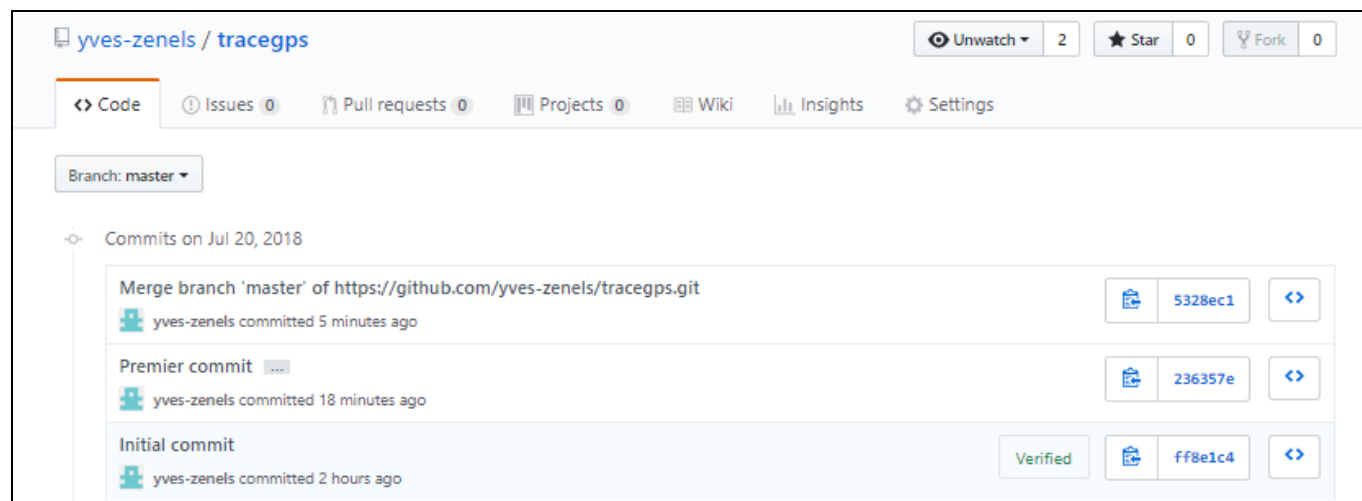
3 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Commit	Time
modele	Premier commit	17 minutes ago
Mobile_Detect.php	Premier commit	17 minutes ago
README.md	Initial commit	2 hours ago
afficher_phpinfo.php	Premier commit	17 minutes ago
essai.php	Premier commit	17 minutes ago
test_accès_bdd_tracegps.php	Premier commit	17 minutes ago

README.md

tracegps



Repository: yves-zenels / tracegps

Unwatch 2 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master

Commits on Jul 20, 2018

Commit	Hash	Link
Merge branch 'master' of https://github.com/yves-zenels/tracegps.git	5328ec1	<>
Premier commit	236357e	<>
Initial commit	ff8e1c4	<>

7- On fait le point ?

Et voilà ! Nous avons interagi pour la première fois avec notre dépôt distant et nous avons indiqué à EGit que la branche **main** sur le dépôt distant correspond à la branche **master** de notre dépôt local.

À partir de maintenant, Eclipse vous indiquera le nombre de commits effectués sur cette branche qui n'ont pas été **pushés** sur le dépôt distant (et inversement si jamais la branche distante contient des commits que vous n'avez pas récupérés en local).

Maintenant, chaque développeur doit se créer un projet local (vide), lui associer un dépôt local, et lier le dépôt local au dépôt distant ; ce sera l'objet du document suivant.

