

06 - Git (esup 4)

Préambule

Cette documentation est complémentaire à la documentation d'installation qui est à destination de l'exploitant.

Elle permet d'installer, configurer, versionner et mettre à jour le package ESUP au travers de **GIT**.

Atlassian propose une documentation Git très bien présentée (en anglais) : <https://www.atlassian.com/git>

Deux tutoriels très détaillés et en Français (www.siteduzero.com et git-scm.com) sont utilisés dans la présente page. Des liens plus précis sont cependant listés.

Voici le lien du projet : <https://github.com/EsupPortail/esup-uportal>

À partir du lien ci-dessus, vous trouverez toutes les informations pour personnaliser les commandes des tutoriels ci-dessous.



Bonnes pratiques

Jasig décrit [ici](#) les bonnes pratiques d'utilisation de Git. On y trouve notamment le [workflow](#) à suivre pour utiliser GIT afin de maintenir son installation locale.

Débutant :

1 - Installer GIT ? | 2 - Régler le proxy (si nécessaire) ? | 3 - Cloner ? | 4 - Passer d'une branche à une autre ? | 5 - Commiter des configurations dans sa branche ? | 6 - Récupérer les dernières mises à jour du portail depuis le github Esup ? | 7 - Faire un merge sur un tag EsupPortail (montée de version sans conflit) ? | 8 - Réaliser un merge en cas de conflit sur un fichier ?

Débutant ++ :

9 - Comparer 2 branches ? | 10 - Consulter l'historique d'un fichier ? | 11 - Faire un tag de la version ? | 13 - Annuler un commit fait par erreur, revenir à une version antérieure de fichier ?

1 - Installer GIT ?

Objectif

Installer les outils nécessaires à Git.

Commande

CentOs

```
sudo yum install git gitk gitg
```

Debian

```
sudo apt-get install git-core gitk gitg
```

<http://windows.github.com/> (Windows)

Résultat attendu

Application cliente installée.

Documentation

www.siteduzero.com - git-scm.com

2 - Régler le proxy (si nécessaire) ?

Objectif

Permettre la communication avec GitHub.com.

Commande

Exécuter Git Shell (Windows)

Unix & Windows

```
git config --global http.proxy http://<LOGIN>:<PASSWORD>@<PROXYADDRESS>:<PORT>
```

Résultat attendu

Pas de blocage de communication de la part du proxy.

Documentation

<https://gist.github.com/336211>

3 - Cloner ?

Cloner : télécharger sur son poste

Objectif

Préparer le déploiement de la nouvelle version dans son établissement en récupérant l'avant dernier tag.

Commande

Unix & Windows

```
git clone https://github.com/EsupPortail/esup-uportal.git
```

Résultat attendu

Sortie de console

```
Cloning into 'esup-uportal'...
remote: Counting objects: 157549, done.
remote: Compressing objects: 100% (42597/42597), done.
remote: Total 157549 (delta 90619), reused 155749 (delta 89147)
Receiving objects: 100% (157549/157549), 129.96 MiB | 1.11 MiB/s, done.
Resolving deltas: 100% (90619/90619), done.
Checking out files: 100% (3263/3263), done
```

Pour connaître la liste des tags disponibles :

Unix & Windows

```
git tag
```

Sortie de console

```
...
trunk-pre-pluto_1.1
umobile-1.0.0
umobile-1.0.0-RC1
umobile-parent-1.0.0-RC1-SNAPSHOT
umobile-server-1.1-RC1
uportal-3.2.5
uportal-4.0.0
uportal-4.0.0-RC1
uportal-4.0.0-RC2
uportal-4.0.1
uportal-4.0.2
uportal-4.0.3
uportal-4.0.4
uportal-4.0.5
uportal-4.0.6
uportal-4.0.6.SR1
uportal-4.0.7
uportal-4.0.8
uportal-4.0.8-esup-0-avant-presta-1
uportal-4.0.11-esup-1
```

Copiez le nom de la version désirée (afin de ne pas récupérer la version en cours de développement) et en faire une branche esup4-univ pour bien distinguer celle-ci.

Pour récupérer la version choisie :

Unix & Windows

```
git checkout -b esup4-univ uportal-4.0.11-esup-1
```

Documentation

www.siteduzero.com - git-scm.com

4 - Passer d'une branche à une autre ?

Objectif

Tester une branche, revenir sur une branche, etc ...

Commandes

Unix & Windows

```
git checkout master
ou
git checkout esup4-univ
```

Résultat attendu

Sortie de console

```
Switched to branch 'master'
Switched to branch 'esup4-univ'
```

Documentation

www.siteduzero.com - git-scm.com

5 - Commiter des configurations dans sa branche ?

Objectif

Effectuer une modification, sauvegarder (en local) les modifications de configurations effectuées.

Exemple avec une modification du fichier esup.properties :

Commandes

Unix & Windows

```
git add <Path>/filters/esup.properties
git commit -m "Changement de connecteur" <Path>/filters/esup.properties
```

Résultat attendu

Sortie de console

```
[esup4-univ a82e27f] Changement de connecteur
1 file changed, 0 insertions(+), 0 deletions(-)
rewrite <Path>/filters/esup.properties(78%)
```

Documentation

www.siteduzero.com - git-scm.com

6 - Récupérer les dernières mises à jour du portail depuis le github Esup ?

Objectif

Mettre à jour le remote (Esup Github) en vue de la mise à jour de la branche.

Commande

Se placer dans le répertoire du projet

Unix & Windows

```
git fetch origin
```

Repérer la dernière version stable et copier son nom (avec gitg par exemple en regardant origin/master) , par exemple uportal-4.0.11-esup-2.

7 - Faire un merge sur un tag EsupPortail (montée de version sans conflit) ?

Objectif

Faire un merge du tag uportal-4.0.11-esup-2 dans sa propre branche.

Conserver la configuration malgré la montée de version.

Commandes :

[Committer](#) les fichiers modifiés avant de faire le merge.

Se placer dans la branche que l'on souhaite mettre à jour

Unix & Windows

```
git checkout ent-monuniv
```

Faire le merge :

Unix & Windows

```
git merge uportal-4.0.11-esup-2
```

Résultat attendu

Sortie de console

```
Updating c6499c1..a82e27f
Fast-forward
 README | Bin 3228 -> 3317 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
```

Documentation

www.siteduzero.com

8 - Réaliser un merge en cas de conflit sur un fichier ?

Objectif

Gérer les erreurs de l'étape 8 ci-dessus.

Commandes

Éditer le fichier conflictuel avec l'éditeur associé. Repérer la section ressemblant à ceci :

Fichier en conflit

```
...
<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
=====
<div id="footer">
  please contact us at support@github.com
</div>
>>>>>> prob53:index.html
...
```

Il faut remplacer l'ensemble de cette section (y compris <<<<<<, ===== et >>>>>>) par ce qui est désiré dans le fichier.

Résultat attendu

Fichier mergé

```
...
<div id="footer">
  please contact us at email.support@github.com
</div>
...
```

Puis faire le commit !

Documentation

git-scm.com

9 - Comparer 2 branches ?

Objectif

Par exemple :

- retrouver les différentes modifications (configuration, look, patch) effectuées par rapport au tag de référence.
- regarder les modifications/différences entre 2 tags
- etc.

Commandes

Consulter les tags et copier les noms des deux tags que l'on souhaite comparer :

Unix & Windows

```
git tag
```

Unix & Windows

```
git diff tag1 tag2
```

Oui si on veut comparer la branche courante avec le tag1 :

Unix & Windows

```
git diff tag1
```

Résultat attendu

```
...
$portalBarFontSize           : 100%;
$portalBarTextAlign          : right;
-$portalBarTextTransform     : capitalize;      # ce qui est supprimé
+$portalBarTextTransform     : none;             # ce qui est ajouté
$portalBarLink               : $cllighter;
$portalBarLinkHover          : $white;
...
```

A noter que des outils graphiques peuvent ici proposer des visions intéressantes de diff ; ce même si git diff propose un certain nombre d'options qui peuvent s'avérer intéressantes également.

Documentation

alx.github.com - git-scm.com

10 - Consulter l'historique d'un fichier ?

Objectifs

Comparer les deux dernières configurations d'un établissement.
Voir ce qui a changé dans le dernier commit.

Commandes

Unix & Windows

```
git log fichier
```

Résultat attendu

```
/home/sweethome/GitHub/esup-uportal [7-SGBD]> git log README
commit b332be2266b4018a677fa2439f34db8e56aef178
Author: Vincent Bonamy <vbonamy@gmail.com>
Date: Mon Jan 21 11:48:06 2013 +0100
```

MAJ README ESUP : esup.properties - simple renommage de propriétés de configuration lda

```
commit fe754564d9562ae40431507d05a6d5354265e83d
Author: Vincent Bonamy <vbonamy@gmail.com>
Date: Wed Dec 12 16:38:11 2012 +0100
```

Update README : git clone with read-only access

```
commit 7b39fedb27a656eac4a6c2b13b57e79dfa3cb80b
Author: Vincent Bonamy <vbonamy@gmail.com>
Date: Fri Dec 7 09:47:36 2012 +0100
```

README -> README-JASIG ; README.esup -> README

```
commit e999d58c61c357fdd19d5c38e7f7c75f6eaa0c4e
Author: Drew Wills <wills.drew@gmail.com>
Date: Tue Oct 2 14:35:04 2012 -0700
```

NOJIRA: Correct outdated info in the README

De même des outils graphiques (github, gitg) peuvent également être intéressants ici lorsqu'on souhaite consulter l'historique des commits ...

Documentation

git-scm.com

11 - Faire un tag de la version ?

Objectif

Marquer une configuration comme fonctionnelle.

Commandes

Unix & Windows

```
git tag -a v1.4 -m 'my version 1.4'
```

Résultat attendu

Aucune sortie. Vérification de la création possible à l'aide de la commande tag.

Documentation

www.siteduzero.com - git-scm.com

13 - Annuler un commit fait par erreur, revenir à une version antérieure de fichier ?

Objectif

En cas de problèmes, revenir à une version fonctionnelle ; défaire un commit ...

Commandes

Il y a plusieurs moyens d'"annuler" un commit, on peut défaire un commit via un commit qui fait les modifications contraires (**revert**) ... et on peut réécrire l'histoire aussi si bien qu'o n'aura plus de trace du commit annulé (**reset**).

Il est plus simple de faire un revert, cela d'autant plus si vous avez déjà partagé le commit avec une (ou plusieurs) autre(s) machine(s) (en fait dans ce cas, il est Très conseillé de ne pas faire de reset).

En plus, même si vous annulez la modification, cela peut être intéressant de conserver cette tentative dans votre historique.

Unix & Windows

```
git revert d6d98923868578a7f
```

```
# Refait un commit qui "défait" le commit d6d98923868578a7f
```