

TP Git

Thomas Clavier

1 configuration

À l'aide de la commande «git config» configurez votre nom (user.name) votre email (user.email) et le serveur mandataire (http.proxy) dans git. L'URL du serveur mandataire de l'université est <http://cache.univ-lille1.fr:3128>

Vérifier à l'aide de la commande «git config -list»

2 L'espace de travail

À partir de là nous allons construire un blog pour le groupe de TP.

2.1 Initialisation

Créez un répertoire de travail et initialisez le comme un dépôt git local.

2.2 Historique de version

Dans ce blog, tous les articles sont rédigés en utilisant le format Markdown (<http://daringfireball.net/projects/markdown/syntax>).

2.2.1 Premier article

Dans l'espace de travail que vous avez initialisé comme dépôt local, créez un sous répertoire «content».

Avec votre voisin choisissez un sujet, puis créez le même article «content/votre-sujet.md» chacun dans votre dépôt local avec le contenu suivant :

```
+++  
date = "2014-01-01T12:00:00+01:00"  
draft = true  
title = "Mon sujet"  
+++
```

```
# Un titre
```

Attention, la date de publication est au format iso 8601.

Vous pouvez par exemple faire un article avec les réponses à ce TP.

À l'aide de «git add», «git commit» et «git log», l'ajouter dans l'index comme fichier à suivre, puis validez le changement. La commande «git log -oneline -decorate -graph -all» devrait vous donner :

```
* a0712c7 (HEAD -> master) Add my first article
```

Observez l'historique des changements, quelles sont les informations disponibles ? À quoi correspondent les champs ?

2.2.2 Second changement

Ajoutez quelques informations dans votre article puis validez un nouveau commit.

Par exemple :

```
+++
date = "2014-01-01T12:00:00+01:00"
draft = true
title = "Mes réponses au TP Git"
+++

# TP Git

## Configuration

    git config --global user.name "Pierre Dupond"
    git config --global user.email "pierre.dupond@univ-lille1.fr"
    git config --global http.proxy http://cache.univ-lille1.fr:3128
```

La commande «git log -oneline -decorate -graph -all» devrait vous donner :

```
* b7f0aaa (HEAD -> master) Add some content in my article
* a0712c7 Add my first article
```

Comment obtenir le diff entre les 2 derniers commits ?

2.2.3 Suppression

- Créez un article «content/lego.md» puis validez un nouveau commit.
- Supprimez l'article «content/lego.md» puis validez un nouveau commit.

La commande «git log -oneline -decorate -graph -all» devrait vous donner :

```
* 8c18094 (HEAD -> master) Remove article about Lego
* c33514e Add article about Lego
* b7f0aaa Add some content in my article
* a0712c7 Add my first article
```

Que voit-on dans l'historique ? Comment, à l'aide de la commande «git checkout» peut-on retrouver le fichier en question ?

La commande «git log -oneline -decorate -graph -all» devrait vous donner :

```
* 8c18094 (master) Remove article about Lego
* c33514e (HEAD) Add article about Lego
* b7f0aaa Add some content in my article
* a0712c7 Add my first article
```

2.2.4 Retour en arrière

Revenir à la situation tel que la commande «git log -oneline -decorate -graph -all» vous donne :

```
* 8c18094 (HEAD -> master) Remove article about Lego
* c33514e Add article about Lego
* b7f0aaa Add some content in my article
* a0712c7 Add my first article
```

Ajoutez quelques informations dans votre article sans valider un nouveau commit.

Comment faire pour annuler le travail et revenir à la dernière version «commité» du fichier ?

La commande «git log -oneline -decorate -graph -all» devrait vous donner :

```
* 8c18094 (HEAD -> master) Remove article about Lego
* c33514e Add article about Lego
* b7f0aaa Add some content in my article
* a0712c7 Add my first article
```

- Créez un article «content/satoshi-tajiri.md» puis validez un nouveau commit.

Comment faire pour annuler ce dernier commit et revenir avec une copie de travail sans ce dernier fichier en HEAD ?

La commande «git log -oneline -decorate -graph -all» devrait vous donner :

```
* 655dc4c (HEAD -> master) Cancel "Add personal details in my article"
* 8d292f3 Add personal details in my article
* 8c18094 Remove article about Lego
* c33514e Add article about Lego
* b7f0aaa Add some content in my article
* a0712c7 Add my first article
```

3 Partager

3.1 Via un dépôt partagé

Indiquons à Git que nous souhaitons échanger des données avec le dépôt distant suivant :

```
http://iut.azae.net/git/tp-git.git
```

Nous appelons ce dépôt «origin».

Puis envoyez l'ensemble des modifications de la branche courante (master) vers le dépôt «origin»

Que se passe-t-il ?

Visitez <http://iut.azae.net/> pour admirer votre travail.

3.2 Via un second dépôt

Aller sur un serveur public comme <https://github.com/>, <https://bitbucket.org/> ou <https://gitlab.com>, une fois connecté créez un nouveau repository : tp-git-blog. Puis ajouter ce nouveau remote sous le nom «kura». Faire un push vers ce dépôt et observer dans l'interface web le résultat.