

# Présentation des outils de coopération

## Formation ANR-Digidoc

Corentin Lallier

*Juin 2012*

## Sommaire

Utilisation de  
Bitbucket

# Utilisation de Bitbucket

Utilisation de  
Mercurial

# Utilisation de Mercurial

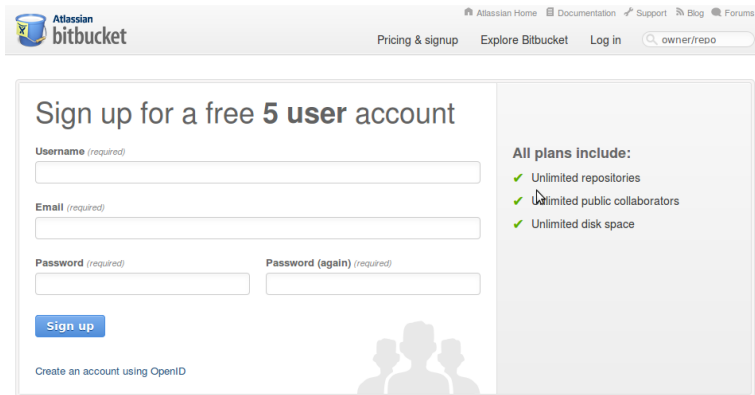
Présentation du  
serveur  
DIGIDOC

# Présentation du serveur DIGIDOC

Conclusion de  
la première  
partie

## Bitbucket

`https://bitbucket.org/account/signup/`



The screenshot shows the Bitbucket sign-up page. At the top, there is a navigation bar with the Bitbucket logo and links for Atlassian Home, Documentation, Support, Blog, and Forums. Below the navigation bar, there are links for Pricing & signup, Explore Bitbucket, and Log in, along with a search bar containing 'owner/repo'. The main content area is titled 'Sign up for a free 5 user account'. It contains a form with the following fields: Username (required), Email (required), Password (required), and Password (again) (required). A blue 'Sign up' button is located below the form. To the right of the form, there is a section titled 'All plans include:' with three bullet points: 'Unlimited repositories', 'Unlimited public collaborators', and 'Unlimited disk space'. At the bottom of the form, there is a link to 'Create an account using OpenID' and a graphic of three stylized human figures.

Atlassian  
bitbucket

Atlassian Home Documentation Support Blog Forums

Pricing & signup Explore Bitbucket Log in owner/repo

### Sign up for a free 5 user account

**Username** (required)

**Email** (required)

**Password** (required) **Password (again)** (required)

[Sign up](#)

[Create an account using OpenID](#)

**All plans include:**

- ✓ Unlimited repositories
- ✓ Unlimited public collaborators
- ✓ Unlimited disk space

## Projet Digidoc

<https://bitbucket.org/clallier/digidoc>

## Présentation rapide

**Wiki** : Home du projet. C'est aussi un wiki complet.

**Overview** : Possibilité de cloner le dépôt, voir les derniers commits, accès au flux RSS, etc ...

**Issues** : gestionnaire de tickets.

## Accès aux sources

**Accès** : via inscription.

**Mail** : identifiant BitBucket + adresse IP

## Principe de base

- ▶ sauvegardes **incrémentales** des modifications / fichiers
- ▶ dépôt **local** VS **global**

## Les commandes

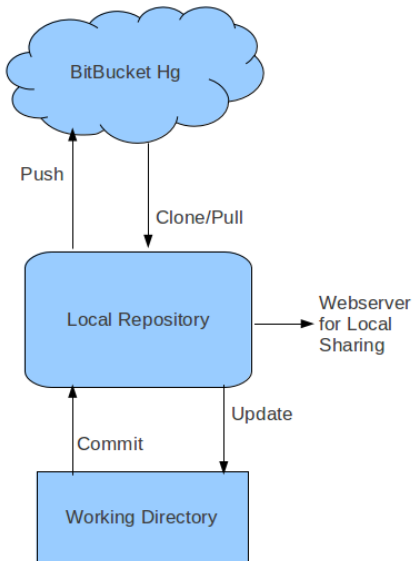
**hg commit** : sauvegarde locales des modifications

**hg push** : sauvegardes globales partagées et incrémentales des modifications

**hg pull** : lister les modifications globales des autres utilisateurs

**hg update** : récupérer les modifications globales des autres utilisateurs

**hg merge** : fusion des modifications locales et globales



## Mercurial

**Lin** : *sudo apt-get install hg*

**Win** : <http://mercurial.selenic.com/wiki/Download>

Choisir le bundle All-In-One : TortoiseHg with Windows Explorer shell integration

## TortoiseHG

Permet de gérer simplement ses différents dépôts avec les commandes usuelles associés.

**Lin** : *sudo apt-get install tortoisehg* (dans les dépôts universe)

**Win** : déjà fournit avec le bundle

## Plan du TP

- ▶ Création du .hgrc
- ▶ Création d'un dépôt de test
- ▶ Création d'un fichier texte
- ▶ Cloner et modifier un autre utilisateur
- ▶ Pull update merge



## Dans le fichier .hgrc

( /home/.hgrc ou c:\Users\XXX\mercurial.ini )

```
[auth]
repo1.prefix = https://bitbucket.org/
repo1.username = NAME
repo1.password = PASSWORD
```

## Créer un dépôt

1. Etre loggé sur BitBucket.
2. dans [Repository](#)
3. 'Create Repository'
4. remplir les champs nécessaires : création du dépôt **global**

## Récupération et modification

1. Se placer dans le dossier de travail (création du dépôt **local**)
2. `hg clone` `https://usr_name@bitbucket.org/creator_name/repo_name`  
(ex : `hg clone https://clallier@bitbucket.org/clallier/digidoc`)
3. Créer un fichier, le modifier, puis l'ajouter via `hg add *`
4. Sauvegarde sur le dépôt local : `hg commit -m "message"`
5. Sauvegarde sur le dépôt global : `hg push`

## Récupération et modification

1. `hg clone` du dépôt de votre voisin.

```
https://usr_name@bitbucket.org/creator_name/repo_name
```

2. Modifier le fichier
3. Sauvegarde sur le dépôt local : `hg commit -m "message"`
4. Sauvegarde sur le dépôt global : `hg push`
5. aller sur la page du projet : les modifications ont été prises en compte.

## Serveur

- ▶ Centralise les ressources
- ▶ Annuaire des algorithmes disponibles

## APIs

- ▶ Différents langages C++, C#. A terme Java, ...
- ▶ Accès aux ressources en lecture/écriture
- ▶ Appel d'algorithmes distants
- ▶ Création et mise à disposition d'algorithmes (présenté dans la seconde partie)

<http://digidoc-repo.labri.fr/DoQuBookWeb/index.jsp>

### Page Collection Browser

Navigate thought and edit your page collections.



Designed and built with [Bootstrap](#), [jQuery](#), and [JSP](#)  
 with love from [DigitalDoc](#) France. Licensed under [CC BY SA](#).  
 © Lallier & Aude - 2012/2013, 2014

[Back to top](#)

### Page editor

Call distant plugins on your page.

590 page-6.png



Designed and built with [Bootstrap](#), [jQuery](#), and [JSP](#)  
 with love from [DigitalDoc](#) France. Licensed under [CC BY SA](#).  
 © Lallier & Aude - 2012/2013, 2014

[Back to top](#)

## REST

- ▶ Protocole basé sur HTTP
- ▶ Représentation hiérarchique des ressources
- ▶ Accessible (ex : utilisation via navigateur, interface web , API c++, etc)
  - ▶ ex : la ressource "page collection" est accessible sur :
  - ▶ `http://digidoc-repo.labri.fr/DoQuBookWeb/rest/pagecollection`  
**liste les collections disponibles**
  - ▶ `http://digidoc-repo.labri.fr/DoQuBookWeb/rest/pagecollection/XXX`  
**accès à la collection XXX**

## Conclusion :

- ▶ Prise en main de Mercurial
- ▶ Prise en main du système de gestion offert par BitBucket
- ▶ Présentation de l'architecture du projet Digidoc :
  - ▶ Serveur / clients
  - ▶ Stockage de documents
  - ▶ Partage d'algorithmes
  - ▶ Utilisation à distance