

Mise en place du système de version

Mise en place du logiciel de versioning

Nous avons tout d'abord du créer notre projet sur google code via l'interface suivante:

Project name	<input type="text" value="ppe2-m2l"/> <small>Example: my-project-name That project name is not available.</small>
Project summary	<input type="text" value="Application de gestion de salle"/>
Description	<input type="text" value="Dans ce projet, nous gérons la locations, suppression et ajout de salle sur le framework web2py"/>
Version control system	<input checked="" type="radio"/> Git <input type="radio"/> Mercurial <input type="radio"/> Subversion
Source code license	<input type="text" value="GNU GPL v3"/>
Project label(s)	<input type="text"/> <input type="text"/> <input type="text"/> add another row
	<input type="button" value="Create project"/>

Après avoir valider, une nouvelle fenêtre apparaît:



The screenshot shows the GitHub project page for 'ppe2-m2l-groupe1'. The page title is 'ppe2-m2l-groupe1' with the subtitle 'Application de gestion de salle'. The navigation bar includes 'Project Home', 'Wiki', 'Issues', 'Source', and 'Administer'. Below the navigation bar, there are tabs for 'Summary' and 'People'. A tip box states: 'Tip: Project owners, see our [Getting Started](#) guide for steps to configure your project.' The 'Project Information' section on the left lists: 'Stared by 0 users', 'Project feeds', 'Code license GNU GPL v3', 'Members SIO.gendreaud.quentin', and 'Your role Owner'. The main content area contains the text: 'Dans ce projet, nous gérons la locations, suppression et ajout de salle sur le framework web2py'. At the bottom, there are links for 'Terms - Privacy - Project Hosting Help' and 'Powered by Google Project Hosting'.

Nous avons donc ensuite ajouter les autres membres du groupe via l'interface suivante:

Configuration du proxy pour git:

```
git config --global http.proxy http://172.30.137.29:3128
git config --global https.proxy https://172.30.137.29:3128
```

Configurer l'utilisateur sur git:

```
git config --global user.name "quentin"
git config --global user.email SIO.gendreaud.quentin@gmail.com
```

Tester si la commande de configuration de proxy :

```
ex : git clone
https://sio.derrien.alexandre@code.google.com/p/ppe2-m2l-groupe1/
```

Commandes GIT de base

Importer un fichier local vers le serveur:

```
git push origin master
```

[Vue suite à la modification du fichier importer:](#)

Application de gestion de salle

Project Home Wiki Issues Source Administer

Repository: default Checkout Browse Changes Clones Request code review

Source path: git/ Download zip | tar.gz

Filename	Size	Rev	Date	Author
test.txt	15 bytes	6060f22af7a6	Today (3 minutes ago)	SIO.gendreaud.quentin <SIO.gendreaud.quentin>

[Create or upload a new file](#)

Importation des fichiers de l'application: Copie des fichiers en local

```
git add * -f
git commit -m "importation des fichiers"
git push
```

Suppression du fichier test.txt:

```
git pull
git rm test.txt
git commit -m "suppression des fichiers texte"
git push
```

Mise en place de l'environnement de test

Créer un environnement de dev sur web2py:

Téléchargement de web2py.

Il faut ensuite décompresser le fichier dans le dossier git (sur mon pc **user/test_git**)

On exécute web2py via web2py.exe

On clic ensuite sur mes sites le mot de passe est celui saisi dans la fenêtre.

On créer ensuite un nouvelle application (chez moi **MRBS_Devs**).

Puis depuis la console git on se rend dans le dossier **web2py/applications/leNomDuProjetCreer**.

Il faut à présent vider le dossier pour le remplacer par la copie. Il faut alors faire : **rm -rf *** .

Ensuite on fait : **git clone**

https://SIO.gendreaud.quentin@code.google.com/p/pppe2-m2l-groupe1/ .

From:

<https://wiki.viper61.fr/> - **Viper61's Wiki**

Permanent link:

https://wiki.viper61.fr/sio/pppe2/mission_3/sys_ver

Last update: **18/09/2016 02:54**