

INSTALL :

- Installation de apache-tomcat :

Dans un premier temps, pour installer le serveur Web Tomcat, nous avons besoin de « Java » sur notre machine virtuelle. Pour ce faire, il suffit d'entrer la ligne de commande suivante :

```
sudo apt-get install openjdk-7-jdk
```

Une fois Java installé, nous procédons à l'installation de Tomcat. Pour ce faire, il suffit d'entrer la ligne de commande suivant :

```
sudo apt-get install tomcat7
```

- Récupération du projet :

Afin de récupérer le projet, il suffit d'entrer la ligne de commande suivante dans le répertoire adéquat :

```
$ git clone git://github.com/MichelMESSAK/ProjectWeb.git
```

- Installation d'Equilatero :

Afin de pouvoir utiliser Equilatero, avec Tomcat, il faut le déplacer le fichier « Equilatero.war » présent dans le dossier « ../Equilatero/dist/ » dans le dossier webapps de Tomcat. Pour ce faire, il suffit d'entrer la ligne de commande suivante :

```
Sudo cp /localisation_of_Equilatero.war var/lib/tomcat7/webapps/
```

Une fois cela effectué, il faut aller dans le répertoire « usr/share/tomcat7 » et crée le répertoire logs avec la commande suivante :

```
Mkdir log
```

Pour finaliser l'installation, il suffit de redémarrer Tomcat par la ligne de commande suivante :

```
Service tomcat7 restart
```

Tomcat est enfin initialisé.

Afin de pouvoir faire fonctionner notre programme, il nous faut également créer une variable d'environnement afin de pouvoir trouver le répertoire de configuration et de log.

Pour ce faire, il suffit de modifier le fichier « catalina.sh » présent dans le répertoire « /usr/share/tomcat7/bin/ ». Modifier le fichier afin d’y inscrire à cette endroit :

```
Export FC_HOME= "path_of_WEB/FC_repository"
```

Pour savoir si la variable s’est bien initialisée, ajouter la ligne suivante entre les autres du même type :

```
Echo "Using FC_HOME: $FC_HOME"
```

Lors de l’initialisation de apache, la ligne “Using FC_HOME : / path_of_WEB/FC ” s’affichera.

Pour la suite, il ne reste plus que la base de données à ajouter.

- Installation de postgres :

Afin de pouvoir utiliser l’application sur notre machine, il faut pour cela installer une base de données correspondante au programme. Pour ce faire, dans un premier temps, nous allons installer postgres.

```
sudo apt-get install postgresql-9.1 postgresql-client-9.1 postgresql-contrib-9.1
```

Nous allons maintenant créer un utilisateur postgres et lui attribuer un mot de passe.

```
sudo su postgres -c psql
ALTER USER postgres WITH PASSWORD 'unsecret';
\q
```

Pour la suite, il va falloir modifier un document. Pour ce faire, entrez la ligne de commande suivante :

```
sudo nano /etc/postgresql/9.1/main/pg_hba.conf
```

Cherchez la ligne

```
# Database administrative login by Unix domain socket
local all postgres peer
```

Et modifiez-la comme ceci :

```
# Database administrative login by Unix domain socket
local all postgres md5
```

Pour que les changements soient pris en compte, il va falloir redémarrer postgres.

```
sudo service postgresql restart
```

Par la suite, nous allons créer une base de données afin de pouvoir utiliser notre application. Pour ce faire, nous devons rentrer les lignes de commande suivante :

```
Sudo su postgres  
Createdb fc
```

Il suffit par la suite de rentrer le mot de passe de l'utilisateur "postgres" et la base de données est créée. Il nous reste maintenant à restaurer cette même base. Pour ce faire, il suffit de rentrer ces lignes de commandes à partir de l'utilisateur « root » :

```
Sudo su postgres -c psql  
\c fc  
\i /path/to/yourbackup.sql
```

Redémarrer postgres SQL et la base de données est maintenant opérationnelle.