Diacamma

Diacamma Syndic est un logiciel libre qui permet de gérer simplement une copropriété

- Diacamma - Clôture de l'exercice en cours et report à nouveau
- Diacamma - mise à jour des modules et sauvegarde des instances
- Installation de Diacamma sur serveur Linux Ubuntu/Debian

Diacamma - Clôture de l'exercice en cours et report à nouveau

> Guide rapide pour clôturer l'exercice en cours, créer un nouvel exercice comptable et effectuer le report à nouveau.

Dans `Comptabilité > Écritures comptables`, contrôler et clôturer toutes les écritures comptables de l'exercice en cours.

Dans `Administration > Configuration comptable`, créer un nouveau plan comptable en important le précédent afin de conserver tous les codes et comptes du plan comptable précédent.

Dans `Comptabilité > Plan comptable` :

1. clôturer l'exercice en cours et choisir comment ventiler le résultat de l'exercice
2. choisir le nouvel exercice créé précédemment et faire le `Report à nouveau`
3. cliquer sur `Commencer` pour démarrer le nouvel exercice

Dans `Administration > Configuration comptable`, activer le nouvel exercice.

Dans `Gestion > Catégories de charges`, reporter et corriger si nécessaire les budgets sur le nouvel exercice.

> Le nouvel exercice comptable est prêt à être utilisé et les appels de fonds peuvent être créés

Diacamma - mise à jour des modules et sauvegarde des instances

Diacamma est fourni avec un script, disponible dans /var/lucterios2 (launch_lucterios.sh), permettant d'effectuer les opérations courantes de maintenance telles que ajouter et modifier des instances, mettre à jour les modules, archiver et restaurer une instance.

Mise à jour des modules

Vérifier la présence de nouvelles versions des modules

```
cd /var/lucterios2
./launch_lucterios.sh check
```

Exemple de sortie de la commande check

```
check list:

lucterios 2.3.0.18073018 => 2.3.0.18073018
lucterios-standard 2.3.0.18070322 => 2.3.0.18070322
lucterios-documents 2.3.0.18070422 => 2.3.0.18070422
lucterios-contacts 2.3.0.18070609 => 2.3.0.18070609
diacamma-syndic 2.3.0.18073020 => 2.3.0.18073020
diacamma-financial 2.3.0.18072813 => 2.3.0.18072813
diacamma-asso 2.3.0.18070322 => 2.3.0.18070322
```

```
=> No upgrade
```

Lister en détails les modules installés

```
./launch_lucterios.sh installed
```

Mettre à jour tous les modules installés

```
./launch_lucterios.sh update
```

Suite à la mise à jour des modules il faut redémarrer gunicorn pour prendre en compte les nouvelles versions

```
sudo killall gunicorn
cd ~
./Diacamma/lancer.sh
```

Script lancer.sh à créer dans /home/username/Diacamma

```
#!/bin/bash

cd /var/lucterios2/

source virtual_for_lucterios/bin/activate

DJANGO_SETTINGS_MODULE='copro.settings' gunicorn lucterios.framework.wsgi --
bind=127.0.0.1:8000 --daemon --timeout 120
```

```
deactivate
```

Rendre le script exécutable

```
chmod +x Diacamma/lancer.sh
```

Sauvegarde des instances

Lister les instances actives

```
./launch_lucterios.sh listing
```

Archiver (sauvegarder) une instance (copro)

```
./launch_lucterios.sh archive -n copro -f save_copro.tar.gz
```

Restaurer la sauvegarde d'une instance (copro)

```
./launch_lucterios.sh restore -n copro -f save_copro.tar.gz
```

`./launch_lucterios.sh --help`

```
Usage:
lucterios_admin.py <check|installed|listing|refreshall|update>
```

```
lucterios_admin.py <add|archive|clear|delete|modif|read|refresh|restore|security> [option]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -n NAME, --name=NAME  Instance name
  -p APPLI, --appli=APPLI
                        Instance application
  -d DATABASE, --database=DATABASE
                        Database configuration 'sqlite', 'MySQL:...' or
                        'PostGreSQL:...'
  -m MODULE, --module=MODULE
                        Modules to add (comma separator)
  -e EXTRA, --extra=EXTRA
                        extra parameters (<name>=value,...).For 'security':
                        'MODE=<M>,PASSWORD=<xxx>'(<M> equals to 0,1 or 2)
  -f FILENAME, --file=FILENAME
                        file name for restor or archive
  -i INSTANCE_PATH, --instance_path=INSTANCE_PATH
                        Directory of instance storage
```

Installation de Diacamma sur serveur Linux Ubuntu/Debian

Installation de Diacamma

Prérequis

Avoir un serveur PostgreSQL installé et configuré avec une base de données créée et vide (DB diacamma).
Installer PostgreSQL et les bibliothèques spécifiques suivantes :

```
sudo apt-get install postgresql python-psycopg2 libpq-dev libssl-dev
```

Créer la base de données

Connexion à l'utilisateur postgres et à l'invite de commande PostgreSQL.

```
sudo -i -u postgres
sudo psql
```

Sur l'invite de commande de PostgreSQL ("postgres=#" est l'invite de commande, il ne faut pas le retaper), création d'un utilisateur "diacamma" et attribution de droits de création de DB.

```
postgres=# CREATE USER diacamma;
postgres=# ALTER ROLE diacamma WITH CREATEDB;
```

Création de la base de données "diacamma" associée à l'utilisateur "diacamma".

```
postgres=# CREATE DATABASE diacamma OWNER diacamma;
```

Attribution d'un mot de passe à l'utilisateur pour qu'il puisse se connecter à la DB.
Bien mettre le mot de passe entre ' '.

```
postgres=# ALTER USER diacamma WITH ENCRYPTED PASSWORD 'mot_de_passe';
```

Quitter l'invite de commande PostgreSQL et vérifier la création de la DB, l'encodage doit être en UTF8.

```
postgres=# \q
psql -l
```

Se déconnecter de l'utilisateur postgres.

```
exit
```

Installation via les scripts pour linux

Télécharger les scripts d'installation / désinstallation depuis l'adresse https://download.sd-libre.fr/Diacamma_setup.tar.gz et décompresser l'archive.

```
wget http://www.sd-libre.fr/download/Diacamma_setup.tar.gz
tar -xf Diacamma_setup.tar.gz
```

Rendre les scripts exécutables.

```
cd Diacamma
chmod +x install.sh uninstall.sh
```

Lancer l'installation de Diacamma en exécutant le script install.sh

```
sudo ./install.sh
```

Installer le driver PostgreSQL pour Django.

```
cd /var/lucterios2/
source virtual_for_lucterios/bin/activate
pip install psycopg2
deactivate
```

Créer et modifier une instance

Pour créer une instance "copro", activer les modules nécessaires et configurer la base de données PostgreSQL, utiliser la commande suivante.
Ne pas oublier de renseigner le mot de passe PostgreSQL.

```
./launch_lucterios.sh add -n copro -m
"lucterios.contacts,lucterios.documents,lucterios.mailing,diacamma.condominium,diacamma.accoun
ting,diacamma.payoff" -p diacamma.syndic -d
"PostGreSQL:name=diacamma,user=diacamma,password=mot_de_passe,host=localhost"
```

Pour corriger une instance existante, remplacer le paramètre "add" par "modif".

Configurer Diacamma en mode serveur

Installer l'outil Gunicorn

```
cd /var/lucterios2/
source virtual_for_lucterios/bin/activate
pip install gunicorn
deactivate
```

Ajouter les options ci-dessous dans le fichier de configuration copro/settings.py pour ne pas avoir une erreur d'hôte non autorisé et l'interface en français.

```
ALLOWED_HOSTS = ['192.168.1.50', 'localhost', '127.0.0.1']
LANGUAGE_CODE = 'fr'
```

Lancer l'instance "copro" en tâche de fond avec les commandes suivantes.

```
cd /var/lucterios2/
source virtual_for_lucterios/bin/activate
DJANGO_SETTINGS_MODULE='copro.settings' gunicorn lucterios.framework.wsgi --
bind=127.0.0.1:8000 --daemon --timeout 120
deactivate
```

L'instance devrait être accessible à l'adresse locale du serveur sur le port 8000.
http://192.168.1.50:8000