

# Guide d'installation de ManageChart

Réalisé le 19 mai 2021 par William Masson (william.masson@inrae.fr)

L'installation a été effectuée en tant que super-utilisateur sur une machine avec la configuration suivante :

- **Ubuntu 20.04.2 LTS**
- **Java 11.0.11**
- **Nginx 1.18.0**
- **Postgres 12.6**
- **PHP 7.4**
- **Composer 2.0.13**

Avant d'installer ManageChart, bien vérifier que PHP, Composer et Java sont installés. D'autres solutions peuvent être utilisées pour la base de données et le serveur (MySQL, Apache, etc)

## ● Installer PHP

Installer PHP (+ certains modules)

```
apt install php-fpm
apt install -y php-{simplexml,intl,mbstring,zip,mysql,pgsql,sqlite3}
```

Vérifier l'installation de PHP

```
php -v
systemctl status php7.4-fpm nginx
```

## ● Installer Composer

Télécharger l'installeur

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
```

Vérifier l'installeur (la clé peut changer : <https://getcomposer.org/download/>)

```
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'756890a4488ce9024fc62c56153228907f1545c228516cbf63f885e036d37e9a59d27d63f46af1d
4d07ee0f76181c7d3') { echo 'Installer verified'; } else { echo 'Installer
corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
```

Lancer l'installeur

```
php composer-setup.php --filename=composer
mv composer /usr/local/bin/composer
```

Supprimer l'installeur

```
php -r "unlink('composer-setup.php');"
```

Vérifier l'installation de Composer

```
composer -V
```

# ● Installer ManageChart

## Cloner le dépôt Git

Depuis le dossier /opt (ou autre, /var/www par exemple) :

```
git clone git://tucuxi.univ-brest.fr/managechart
```

ou

```
git clone -b devel git://tucuxi.univ-brest.fr/managechart (pour avoir les dernières fonctionnalités)
```

## Installer les dépendances

```
cd managechart
```

```
rm composer.lock
```

```
composer install
```

Il va y avoir une erreur, c'est normal.

## Modifier le fichier UnitOfWork

Dans le fichier vendor/doctrine/orm/lib/Doctrine/ORM/UnitOfWork.php, modifier les continue des lignes 2 636 et 2 665 en continue 2.

```
composer install
```

## Modifier le fichier PostgreSQLSchemaManager.php

Dans le fichier vendor/doctrine/dbal/lib/Doctrine/DBAL/Schema/PostgreSQLSchemaManager.php, changer la fonction \_getPortableSequenceDefinition() par celle-ci :

```
protected function _getPortableSequenceDefinition($sequence)
{
    if ($sequence['schemaname'] != 'public') {
        $sequenceName = $sequence['schemaname'] . "." . $sequence['relname'];
    } else {
        $sequenceName = $sequence['relname'];
    }

    $version = floatval($this->_conn->getWrappedConnection()->getServerVersion());

    if ($version >= 10) {
        $data = $this->_conn->fetchAll('SELECT min_value, increment_by FROM pg_sequences
WHERE schemaname = \'public\' AND sequencename = \'$this->_conn->quote($sequenceName)');
    } else {
        $data = $this->_conn->fetchAll('SELECT min_value, increment_by FROM ' . $this->_platform->quoteIdentifier($sequenceName));
    }

    return new Sequence($sequenceName, $data[0]['increment_by'], $data[0]['min_value']);
}
```

## Changer les droits sur le cache et les logs

```
chown www-data:www-data -R app/cache app/logs
```

## Créer le dossier privateKey

```
mkdir src/Mc/EncryptBundle/Resources/privateKey
```

```
chown www-data:www-data -R src/Mc/EncryptBundle/Resources/privateKey
```

## Compiler les fichiers CSS et JS (nécessite Java)

```
php app/console assets:install web/ --symlink
```

```
php app/console assetic:dump --env=prod
```

## Créer la base de données (nécessite PostgreSQL)

```
sudo -u postgres psql
create database db_managechart_prod;
create user u_managechart_prod with encrypted password 'XXXXXXXXXXXXXXXXXX';
grant all privileges on database db_managechart_prod to u_managechart_prod;
alter database db_managechart_prod owner to u_managechart_prod;
\q
```

## Modifier le fichier parameters.yml

Dans le fichier `app/config/parameters.yml`, modifier les lignes `database_name`, `database_user` et `database_password` avec les informations de l'étape précédente.

## Créer les tables de la base de données

```
php app/console doctrine:schema:update --dump-sql (pour voir les modifications)
```

Si la commande précédente ne marche pas, modifier la valeur `127.0.0.1` de la ligne `database_host` du fichier `app/config/parameters.yml` par la valeur `localhost`

```
php app/console doctrine:schema:update --force (pour appliquer les modifications)
```

## Créer un compte administrateur

```
php app/console fos:user:create
```

```
php app/console fos:user:promote avec le rôle ROLE_SUPER_ADMIN
```

## Créer les fichiers serveur (nécessite Nginx)

```
nano /etc/nginx/sites-available/managechart.conf
```

```
ln -fs /etc/nginx/sites-available/managechart.conf /etc/nginx/sites-enabled
```

## Exemple de fichier de configuration :

```
server {
    listen 8888;
    listen [::]:8888;

    server_name XXXXXXXXXXXXX;
    root /opt/managechart/web;

    location / {
        try_files $uri /app.php$is_args$args;
    }

    location ~ ^/app\.php(/|$) {
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        fastcgi_split_path_info ^(.+\.(php))(/.*)$;
        include snippets/fastcgi-php.conf;

        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        fastcgi_param DOCUMENT_ROOT $realpath_root;

        internal;
    }

    location ~ /\.php$ {
        return 404;
    }

    access_log /var/log/nginx/XXXXXXXXXXXXX_access.log;
    error_log /var/log/nginx/XXXXXXXXXXXXX_error.log;
}
```