ACCESSI DYS

AccessiDys

Installation guide

Version 2.3

Summary

List of tables

1 INTRODUCTION

1.1 AIM OF THE DOCUMENT

The document describes how to install the application for development, qualification or production purpose.

It details elements, procedures and requirements for the installation of the application on a linux 64bits platform.

The Accessidys application is dependent of the “mean.io” framework which is compatible with lots of server configurations.

For complete details of server configurations and specific installations, see : http://learn.mean.io/#mean-stack-prerequisite-technologies.

1.2 SERVEUR

The installation has been tested on:

CentOS server (CentOS Linux release 7.1.1503 (Core) 64bits)

1.3 REFERENCE DOCUMENTATION

N°	Version	Date	Document title	Detail
1	V1.0	18/05/2016	ACCESSIDYS_DAT_ENG_V1.0	Technical architecture design

Table 1-1 : reference documents

1.4 GLOSSARY

Acronyms	Definition
MEAN	Architecture based upon mongoDB, express, AngularJS, NodeJS technologies

Table 1-2 : Acronyms description

2 DESCRIPTION

2.1 REQUIRED SOFTWARES

The application is based on the MEAN framework (http://mean.io/#!/).

Therefore all softwares required for this framework must be installed :

- NodeJS
- MongoDB
- Git

2.2 THE INSTALLATION IS DESCRIBED IN THE CHAPTER : 3.1 PRE-REQUISITES

Execute the command lines below to define specific variables for the installation:

Note : You can update the paths depending of your own configuration.

```
export HOME_SOFT=/home/$USER/Accessidys/softwares
export HOME_APPLI=/home/$USER/Accessidys/application
export HOME_ENV=/home/$USER/Accessidys/env
export HOME_DATA=/home/$USER/Accessidys/data
export PATH="${PATH}:${HOME_SOFT}/node-v0.12.7-linux-x64/bin/"
export PATH="${PATH}:${HOME_SOFT}/mongodb-linux-x86_64-rhel70-3.0.6/bin/"

#creation of directories if not already existed
mkdir -p $HOME_SOFT
mkdir -p $HOME_APPLI
mkdir -p $HOME_ENV
mkdir -p $HOME_DATA
```

Execute the command lines below depending the kind of the installation environment:

Development platform

`export NODE_ENV=dev`
`export HOME_APP=$HOME_APPLI/accessidys`
`export CONF_FILE_NAME=config.json`

Qualification platform

`export NODE_ENV=recette`
`export HOME_APP=$HOME_APPLI/accessidys/dist`
`export CONF_FILE_NAME=config.recette.json`

Production platform
`export NODE_ENV=prod`

```
export HOME_APP=$HOME_APPLI/accessidys/dist
export CONF_FILE_NAME=config.prod.json
```

Required softwares for the installation.

2.3 INSTALLATION DIRECTORY STRUCTURE

There are 4 types of installation :

- Required software
- Application with logs
- Environment configuration for application
- Data of database

Those types will be defined in different paths of the same directory that can be set on different hard drive like for production.

The structure proposed is :

- HOME_SOFT : /home/$USER/Accessidys/softwares/
- HOME_APPLI : /home/$USER/Accessidys/application/
- HOME_ENV : /home/$USER/Accessidys/env/
- HOME_DATA : /home/$USER/Accessidys/data/

The application tree is the following :

HOME_APPLI :

sslcert : folder with HTTPS certificates

logs : logs folder

accessidys : application code

3 INSTALLATION

3.1 PRE-REQUISITES

Execute the command lines below to define specific variables for the installation:

Note : You can update the paths depending of your own configuration.

```
export HOME_SOFT=/home/$USER/Accessidys/softwares
export HOME_APPLI=/home/$USER/Accessidys/application
export HOME_ENV=/home/$USER/Accessidys/env
export HOME_DATA=/home/$USER/Accessidys/data
export PATH="${PATH}:${HOME_SOFT}/node-v0.12.7-linux-x64/bin/"
export PATH="${PATH}:${HOME_SOFT}/mongodb-linux-x86_64-rhel70-3.0.6/bin/"

#creation of directories if not already existed
mkdir -p $HOME_SOFT
mkdir -p $HOME_APPLI
mkdir -p $HOME_ENV
mkdir -p $HOME_DATA
```

Execute the command lines below depending the kind of the installation environment:

Development platform

`export NODE_ENV=dev`
`export HOME_APP=$HOME_APPLI/accessidys`
`export CONF_FILE_NAME=config.json`

Qualification platform

`export NODE_ENV=recette`
`export HOME_APP=$HOME_APPLI/accessidys/dist`
`export CONF_FILE_NAME=config.recette.json`

Production platform

`export NODE_ENV=prod`
`export HOME_APP=$HOME_APPLI/accessidys/dist`
`export CONF_FILE_NAME=config.prod.json`

3.2 REQUIRED SOFTWARES FOR THE INSTALLATION

3.2.1 NodeJS

NodeJS must be installed:

```
cd $HOME_SOFT

wget https://nodejs.org/dist/v0.12.7/node-v0.12.7-linux-x64.tar.gz
tar xvzf node-v0.12.7-linux-x64.tar.gz
cd node-v0.12.7-linux-x64/bin
./npm install -g grunt-cli
./npm install -g yo
./npm install -g bower
```

On any issues see : https://nodejs.org/en/download/package-manager/

3.2.2 MongoDB

MongoDB must be installed:

```
#Installation mongodb
cd $HOME_SOFT
wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-3.0.6.tgz
tar xvzf mongodb-linux-x86_64-rhel70-3.0.6.tgz
```

On any issues see : https://docs.mongodb.com/manual/administration/install-on-linux/

3.2.3 GIT

GIT must be installed:

For Fedora/CentOS/RedHat release of linux :

```
cd $HOME_SOFT

yum install git
```

For Debian/Ubuntu release of linux :

```
cd $HOME_SOFT

apt-get install git
```

3.3 APPLICATION INSTALLATION

The application is cloned from GIT :

```
cd $HOME_APPLI

git clone -b stable https://github.com/AccessiDys/accessidys.git
```

The specific environment directory needs to be created:

```
cd $HOME_APPLI

mkdir logs
mkdir sslcert

cp accessidys/env/config.json $HOME_ENV/$CONF_FILE_NAME
```

For development and qualification environment, you can define a self-signed certificate for the application (Note that it is not safe):

```
cd $HOME_APPLI

cd sslcert
openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem

#Example of data to set :
#Country Name (2 letter code) [XX]:FR
#State or Province Name (full name) []:France
#Locality Name (eg, city) [Default City]:Rennes
#Organization Name (eg, company) [Default Company Ltd]:ORGANIZATION
#Organizational Unit Name (eg, section) []:OR
#Common Name (eg, your name or your server's hostname) []:MyHostName
#Email Address []:
```

For production you need a Signed certificate, and put it in sslcert (cert.pem)

Then the application can be compiled:

```
cd $HOME_APPLI/accessidys
npm install
```

3.4 CONFIGURATION

3.4.1 Application Configuration

The application configuration is based on the config.json file in the env folder.

This file provides :

- The environment name (dev, int, etc.)
- The host of the mongo database (the database port is 27017)
- The database name
- The application URL (in order to be able to call Rest services)
- HTTPS certificate name
- Dropbox API keys (See : 3.4.2 Dropbox API Configuration)
- Email configuration
- Catalog name

```
cd $HOME_ENV
vi $CONF_FILE_NAME
```

config.json configuration file example

```
{
        "NODE_ENV": "dev",
        "MONGO_URI": "localhost",
        "MONGO_DB": "adaptation",
        "URL_REQUEST": "https://localhost:3000",
        "SSL_KEY":"key.pem",
        "SSL_CERT":"cert.pem",
        "DROPBOX_CLIENT_ID": "XXXXX",
        "DROPBOX_CLIENT_SECRET": "XXXX",
        "DROPBOX_TYPE": "sandbox",
        "EMAIL_HOST": "smtp.gmail.com",
        "EMAIL_HOST_UID": "test@gmail.com",
        "EMAIL_HOST_PWD": "xxxx",
        "CATALOGUE_NAME":"adaptation.html"
}
```

3.4.2 Dropbox API Configuration

To allow Accessidys access to dropbox documents, you need to create and configure an API Dropbox application.

Connect on https://www.dropbox.com/developers with your login/Password

- Go to “My Apps”
- Click an “Create app” button
- Choose your API option
- Choose “App folder” type
- Give your app name : AccessidysAppForTest (for example)
- Click on “Create App” button

- Get the App key and App secret to put in the config.json file of the application (see 3.4.1 Application Configuration)

- Add a Redirect URIs in the “OAuth2” section :
  - For dev plateform : https://localhost:3000/auth/dropbox/callback
  - For other platform : https://<HOST_NAME>:<HOST_PORT>/auth/dropbox/callback with HOST_NAME and HOST_PORT to be set with the good value.

3.4.3 Apply Configuration

In order to apply the configuration run the command below :

```
cd $HOME_APPLI/accessidys
grunt build-${NODE_ENV}
```

This command creates files from templates with the right configuration parameters.

New compilation is need to take into account configuration

```
cd $HOME_APP
npm install
```

3.5 APPLICATION MANAGEMENT

3.5.1 Database startup

Database must be started before the application:

```
cd $HOME_DATA
mkdir -p db

nohup mongod -dbpath $HOME_DATA/db >/dev/null 2>&1 &
```

3.5.2 Start Application

Starting up the application in background:

```
cd $HOME_APP
nohup grunt server >/dev/null 2>&1 &
```

3.5.3 Stop Application

When needed, you can stop the application by applying next command lines :

```
#Get process PID
ps -aef | grep grunt
#Example :  node     14530     1  4 Jul13 ?        16:35:18 grunt
#14530 is the PID
#Stop process (remplace <PID> with previous value):
kill <PID>
```

3.5.4 Define an admin account

Start mongo using command lines :

```
mongo
#connecting to: test
#>
use adaptation
#switched to db adaptation
#>
```

Select user account with the email value (remplace **xxxx@xxx.xxx** with user email):

```
db.getCollection('users').find({'local.email':'xxxx@xxx.xxx'})
#{ "__v" : 0, "_id" : ObjectId("535f74ecbea537672da56636"), "dropbox" : { "accessToken" :
"puwOBDrUMwcAAAAAAAAAIUrEhOFasZ27z_SwxaambYysL0npcD6EZqAqujPzOqJ-", "country" : "MA",
"display_name" : "adaptation cned", "emails" : "adaptation@cned.fr", "referral_link" :
"https://db.tt/IWsiOuo2", "uid" : "283269389" }, "local" : { "authorisations" : { "audio"
: true, "ocr" : true }, "email" : "adaptation@cned.fr", "nom" : "cned", "password" :
"33159f1cf13a115084d2f73eec39b705", "prenom" : "adaptation", "role" : "user", "token" :
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJjaGFpbmUiOiJsdWx0MDUyOSJ9.Ll7MbY_F09lD7TTCrWIcSH
U4fATuEj0y2R8N04cwjJ8", "tokenTime" : 1469664239642 } }
```

Update role to admin (remplace **xxxx@xxx.xxx** with user email):

```
db.getCollection('users').update({'local.email':'xxxx@xxx.xxx'}, {
$set:{'local.role':'admin'}})
```

Check update (remplace **xxxx@xxx.xxx** with user email):

```
db.getCollection('users').find({'local.email':'xxxx@xxx.xxx'})
#{ "__v" : 0, "_id" : ObjectId("535f74ecbea537672da56636"), "dropbox" : { "accessToken" :
"puwOBDrUMwcAAAAAAAAAIUrEhOFasZ27z_SwxaambYysL0npcD6EZqAqujPzOqJ-", "country" : "MA",
"display_name" : "adaptation cned", "emails" : "adaptation@cned.fr", "referral_link" :
"https://db.tt/IWsiOuo2", "uid" : "283269389" }, "local" : { "authorisations" : { "audio"
: true, "ocr" : true }, "email" : "adaptation@cned.fr", "nom" : "cned", "password" :
"33159f1cf13a115084d2f73eec39b705", "prenom" : "adaptation", "role" : "admin", "token" :
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJjaGFpbmUiOiJsdWx0MDUyOSJ9.Ll7MbY_F09lD7TTCrWIcSH
U4fATuEj0y2R8N04cwjJ8", "tokenTime" : 1469664239642 } }
```