ACCESSI DYS

AccessiDys

Installation guide

Version 1.0

Summary

List of tables

1 INTRODUCTION

1.1 AIM OF THE DOCUMENT

The document describes how to install the application. It details elements, procedures and requirements for the installation of the application on CentOS.

The aim of the document is to provide required information to the installation team.

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 INSTALLATION

2.1 REQUIRED SOFTWARES FOR THE INSTALLATION

This section described sofwares that need to be installed on the server.

The application has been developed in the MEAN architecture way (http://mean.io/#!/). Therefore all softwares required for this type of architecture must be installed.

The installation procedure of these tools are available on their website. The application does not need specific configuration. The standard one can also be used.

2.1.1 NodeJS

NodeJS must be installed on the environment and added in the user PATH environment variable.

NodeJS is a platform to execute Javascript applications on server side.

Installation :

```
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
```

2.1.2 MongoDB

MongoDB must be installed on the environment and added in the user PATH environment variable.

MongoDB is a document oriented database.

Installation :

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

2.1.3 Tesseract

Tesseract must be installed on the environment and added in the user PATH environment variable.

Tesseract is an OCR engine.

Installation :

```
#Installation ghostscript
yum install ghostscript
```

```
#Installation automake
yum install automake

#Installation libtool
yum install libtool

#installation leptonica
cd ~
mkdir leptonica
cd leptonica
wget http://www.leptonica.com/source/leptonica-1.72.tar.gz
tar -zxvf leptonica-1.72.tar.gz
cd leptonica-1.72
./configure
make
make install

#installation tesseract
cd ~
mkdir tesseract
cd tesseract
wget http://tesseract-ocr.googlecode.com/files/tesseract-ocr-3.02.02.tar.gz
tar -zxvf tesseract-ocr-3.02.02.tar.gz
cd tesseract-ocr
./autogen.sh
./configure
make
make install
ldconfig

#installation de la langue FR
cd /usr/local/share/tessdata
sudo wget https://tesseract-ocr.googlecode.com/files/tesseract-ocr-3.02.fra.tar.gz
gunzip -f tesseract-ocr-*.tar.gz
rm -f tesseract-ocr-*.tar.gz
ls
tar -xf tesseract-ocr-*.fra.tar
rm -f tesseract-ocr-*.fra.tar
mv tesseract-ocr/tessdata/* .
rm -rf tesseract-ocr/
```

2.1.4 ImageMagick

ImageMagick must be installed on the environment and added in the user PATH environment variable.

ImageMagick can perform image optimization to improve the results of the OCR engine.

Installation :

```
wget http://www.imagemagick.org/download/ImageMagick.tar.gz
tar xvzf ImageMagick.tar.gz
rm -f ImageMagick.tar.gz
cd ImageMagick-*/
```

```
./configure
make
make install
ldconfig /usr/local/lib
cd ../
ls
rm -rf ImageMagick-*/
```

2.1.5 ESpeak

ESpeak must be installed on the environment and added in the user PATH environment variable.

ESpeak is a library providing voice synthesis.

Installation :

```
# Installation espeak
yum install espeak

#Installation mbrola (language pack)
cd ~
wget http://www.tcts.fpms.ac.be/synthesis/mbrola/bin/pclinux/mbr301h.zip
cp mbrola-linux-i386 /usr/bin/mbrola
wget http://www.tcts.fpms.ac.be/synthesis/mbrola/dba/fr1/fr1-990204.zip
unzip fr1-990204.zip
cp fr1/fr1 /usr/share/espeak-data/mbrola/fr1
wget http://www.tcts.fpms.ac.be/synthesis/mbrola/dba/fr4/fr4-990521.zip
unzip fr4-990521.zip
cp fr4 /usr/share/espeak-data/mbrola/fr4
```

2.1.6 GIT

GIT must be installed on the environment and added in the user PATH environment variable.

GIT is a tool providing source code management of GIT repositories.

Installation :

```
#Installation git
yum install git
```

2.2 HOW TO INSTALL

The installation of the application is done by cloning the GIT repository and applying the configuration adapted to the environment. Then the application can be started.

2.3 DIRECTORIES LAYOUT

The application must be cloned from GIT to a directory HOME_APPLI/git with HOME_APPLI the root directory which will contains all the application files.

The application tree is the following :

HOME_APPLI : root folder

git :

sslcert : folder with HTTPS certificates

logs : logs folder

accessidys : application code

env : folder containing the config.json specific to the environment

GIT cloning and folders tree creation :

```
mkdir -p CNED/git
cd CNED/git
git clone https://github.com/cnedDI/accessidys.git
cd ..
mkdir logs
mkdir sslcert
cd sslcert
openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem
```

The configuration file config.json must be created in the env directory.

2.4 NODEJS MODULES INSTALLATION

The applications uses NodeJS modules to generate code, execute unit tests and so on…

The following modules are the mainly used by the application :

- Npm : NodeJS modules manager
- Grunt : task manager (code generation, unit tests execution, application startup)
- Karma : unit tests framework
- Express : Rest services framework

Module installation is done by executing the command below in the accessidys folder :

```
npm install
```

2.5 CONFIGURATION GENERATION

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
- Email configuration
- Catalog 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": "dlyns2fma4grgmq",
      "DROPBOX_CLIENT_SECRET": "wnt0y8sjdcm0pca",
      "DROPBOX_TYPE": "sandbox",
      "EMAIL_HOST": "smtp.gmail.com",
      "EMAIL_HOST_UID": "test@gmail.com",
      "EMAIL_HOST_PWD": "xxxx",
      "CATALOGUE_NAME":"adaptation.html"
}
```

In order to apply the configuration run the command below :

```
grunt build-dev
```

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

2.6 APPLICATION STARTUP

Database must be started before the application:

```
mongod –dbpath data/db
```

With data/db the folder where data will be stored.

To start the application, run the command below in accessidys folder:

```
grunt server
```

The application can be accessed at **https://<host>:3000/** where <host> is the server name.

2.7 INSTALLATION FOR DEVELOPMENT

The installation on a developpment platform is identical to the server one. The configuration file must be updated according to the environment.