Table of Contents

[1 – INSTALL PROJECT DEPENDENCIES/REQUIREMENTS 2](#__RefHeading___Toc1100_3660483187)

[1.1 Install Python2 and Virtualenvironment 2](#__RefHeading___Toc2794_3660483187)

[1.2 Install Node Angular 2](#__RefHeading___Toc2796_3660483187)

[1.3 Install Postgres 10 2](#__RefHeading___Toc2798_3660483187)

[1.4 Install Postgres 10 and Extension dependencies for Alt-Project 3](#__RefHeading___Toc2800_3660483187)

[1.5 Bitbucket Clone 3](#__RefHeading___Toc2831_3660483187)

[1.6 Rabbitmq setup 3](#__RefHeading___Toc2802_3660483187)

[1.7 Redis server 4](#__RefHeading___Toc2821_3660483187)

[1.8 Elasticsearch 4](#__RefHeading___Toc2823_3660483187)

[1.9 Cronjobs 5](#__RefHeading___Toc2825_3660483187)

# 1 – INSTALL PROJECT DEPENDENCIES/REQUIREMENTS

### 1.1 Install Python2 and Virtualenvironment

* sudo apt update
* sudo apt install python
* sudo apt install python-pip
* sudo -H pip install virtualenv

Install Python 2 and virtual environment to install python packages/reuirements from dce\_requirements.txt

### 1.2 Install Node Angular

* sudo apt install npm
* sudo npm cache clean -f
* sudo npm install -g n
* sudo n stable
* node –version
* sudo npm install -g @angular/cli@8.3.12

### 1.3 Install Postgres 10

* wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
* sudo apt-get update
* sudo apt-get install postgresql-10

Connect to PostgreSQL

* sudo su - postgres
* psql

Change the password of postgres role

* \password postgres

Change listen address in postgresql.conf to \* if on server to access it remotely

* sudo nano /etc/postgresql/10/main/postgresql.conf

In pg\_hba.conf if on server open connection for remote access by adding the following in:-

* sudo nano /etc/postgresql/10/main/pg\_hba.conf
* host all all 0.0.0.0/0 md5

now restart the server and check it’s status.

* sudo service postgresql restart
* sudo service postgresql status

refer link : <https://yallalabs.com/linux/how-to-install-and-use-postgresql-10-on-ubuntu-16-04/>

### 1.4 Install Postgres 10 and Extension dependencies for Alt-Project

* sudo apt install python3-pip
* sudo pip3 install businesstimedelta==0.2
* sudo apt-get install postgresql-plpython3-10

### 1.5 Bitbucket Clone

* Clone the project from repo and make virtual environment
* Inside the virtualenvironment install dependencies from dce\_requirements.txt
* Run Python Setup.py develop
* For Leggero\_Config.xml and env.service.ts (refer attached or ask from Taukir for latest/updated)
* Path for Leggero\_Config.xml => dce/DCE/commons/Leggero\_Config.xml
* Path for env.service.ts => dce/DCE/appserver/dce/src/app/dce-service/ env.service.ts

### 1.6 Rabbitmq setup

• sudo apt-get install rabbitmq-server

refer link : <https://computingforgeeks.com/how-to-install-latest-rabbitmq-server-on-ubuntu-linux/>

If running multiple celeries on the same server through raabitmq use the following

1. sudo rabbitmqctl add\_user alt\_del\_demo alt\_del\_demo
2. sudo rabbitmqctl add\_vhost alt\_del\_demo\_host
3. sudo rabbitmqctl set\_permissions -p alt\_del\_demo\_host alt\_del\_demo ".\*" ".\*" ".\*"
4. sudo rabbitmqctl set\_user\_tags alt\_del\_demo alt\_del\_demo\_user

and add/update the following in Leggero\_Config.

<CELERY>

<broker>amqp://alt\_del\_demo:alt\_del\_demo@0.0.0.0:5672/alt\_del\_demo\_host</broker>

<backend>amqp://0.0.0.0//</backend>

</CELERY>

### 1.7 Redis server

* sudo apt update
* sudo apt install redis-server
* sudo /etc/init.d/redis-server start
* Check password for Redis in Alt-Del

### 1.8 Elasticsearch

* Download elasticsearch 7.8 tar file from https://www.elastic.co/downloads/past-releases/elasticsearch-7-8-0
* extract the downloaded folder
* open terminal in extracted folder and go to elasticsearch-7.8.0/bin
* type ./elasticsearch example: elasticsearch-7.8.0/bin> ./elasticsearch your elastic search running on <http://127.0.0.1:9200/>
* pip install elasticsearch (In project environment)

**Leggero Configuration**

<ES\_SEARCH>

<HOST>http://localhost:9200/</HOST>

<ES\_USER>elastic</ES\_USER>

<ES\_PASSWORD>elastic</ES\_PASSWORD>

<BIZ\_ORG\_INDEX\_NAME>biz\_org\_idx</BIZ\_ORG\_INDEX\_NAME>

<INDIVIDUALS\_INDEX\_NAME>individuals\_idx</INDIVIDUALS\_INDEX\_NAME>

<CONTENT\_BASE\_INDEX\_NAME>content\_base\_idx</CONTENT\_BASE\_INDEX\_NAME>

</ES\_SEARCH>

**Load Data to ES**

go to the following path in alt-del project **DCE/create\_script/search\_indexes\_data/**

andrun the folllowing scripts.

Load Bizorg\_data to ES

* python bizorg\_search\_insert.py

Load Individual\_data to ES

* python load\_individuals\_to\_es.py

Load content\_base\_data to ES

* python load\_content\_base\_to\_es.py

**For SERVER Enable X-pact security and allow remote access**

Go to elasticsearch-7.8.0/config and edit **elasticsearch.yml**

* xpack.security.enabled: true

restart the server and run below command in bin of elasticsearch

* elasticsearch-setup-passwords interactive

and set the password for elasticsearch

For remote access just add the following in **elasticsearch.yml**

* network.host: 0.0.0.0
* transport.host: localhost
* http.port: 9200
* transport.tcp.port: 9300

### 1.9 Cronjobs

3 cronjobs exist in alt-project as of today

1. Sequence Counter Reset (To reset the sequence for cases).
2. Refresh Case Material View (material view used for reporting) :this cron only for cases\_all\_logged\_mat\_view(ONLY) . Every night run at 01.00.00 (IST)
3. Cases Resolved (Auto Close cases that have been in resolved state for 3 days) **ever day run:**

**1. Sequence Counter Reset**

**#!/bin/bash   
  
echo Current Date and Time: `date +"%Y-%m-%d %T"`   
  
echo "Reseting sequence tracker ..."   
  
/home/aks33/Alt-New-Test/venv/bin/python2 /home/aks33/Alt-New-Test/dce/DCE/dbserver/sequence\_tracker\_reset.py   
  
echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"   
echo ""**Add this data in file **reset\_sequence\_tracker.sh** and schedule it via crontab

0 0 \* \* \* bash /home/aks33/CronScripts/sequence\_reset/reset\_sequence\_tracker.sh >> /home/aks33/CronScripts/sequence\_reset/reset\_sequence\_tracker\_out.txt

**2. Refresh Case Material View**

**#!/bin/bash   
  
echo Current Date and Time: `date +"%Y-%m-%d %T"`   
  
echo "Refreshing material view cases ..."  
  
/home/aks33/Alt-New-Test/venv/bin/python2 /home/aks33/Alt-New-Test/dce/DCE/job\_loads/logged\_cases\_mat\_view.py  
  
echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"   
echo ""**Add this data in file **refresh\_cases\_mat\_view.sh** and schedule it via crontab

1 0 \* \* \* bash /home/aks33/CronScripts/refresh\_cases\_mat\_view/refresh\_cases\_mat\_view.sh >> /home/aks33/CronScripts/refresh\_cases\_mat\_view/refresh\_cases\_mat\_view\_out.txt

**3. Cases Resolved**

**#!/bin/bash   
  
echo Current Date and Time: `date +"%Y-%m-%d %T"`   
  
echo "Resolving cases ..."   
  
/home/aks33/Alt-New-Test/venv/bin/python2 /home/aks33/Alt-New-Test/dce/DCE/job\_loads/case\_resolved\_daily.py  
  
echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"   
echo ""**Add this data in file **cases\_resolved\_daily.sh** and schedule it via crontab

0 3 \* \* \* bash /home/aks33/CronScripts/case\_resolved\_daily/cases\_resolved\_daily.sh >> /home/aks33/CronScripts/case\_resolved\_daily/cases\_resolved\_daily\_out.txt