

# mTrack Set-up Documentation

Samuel Sekiwere

May 15, 2013

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Core Applications</b>	<b>4</b>
<b>3</b>	<b>Virtual Environments</b>	<b>5</b>
<b>4</b>	<b>Dependencies installed Using Pip</b>	<b>6</b>
<b>5</b>	<b>Setup Process</b>	<b>8</b>
5.1	Clone mTrac git repository . . . . .	8
5.2	Add Logging Directories . . . . .	8
5.3	Running mTrack . . . . .	8

# **1 Introduction**

This document describes a full mTrac Set-up onto a new server that will be hosted at the Ministry of Health.

The subsequent sections describe the core applications that are required before you can install mTrac, including the procedures and configurations used to achieve a working set-up.

## 2 Core Applications

The server is currently running Ubuntu 12.10 as the underlying operating system. The following table lists the prerequisite core applications that must be installed on the server before one can attempt to fully set up mTrac:

#	Package	Description
1	build-essential	Informational list of build-essential packages
2	postgresql-9.2	Underlying database management system
3	pip	A tool for installing and managing Python packages
4	python-dev	Header files and a static library for Python
5	git	Fast, scalable, distributed revision control system
6	kannel	WAP and SMS gateway
7	nginx	Small, powerful, scalable web/proxy server
8	supervisor	A System for controlling process state
9	pgbouncer	Lightweight connection pooler for PostgreSQL
10	rabbitmq-server	AMQP server written in Erlang

Table 1: Prerequisite Core Applications

The snippet below shows the commands to run in order to install the prerequisites (on Ubuntu).

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:pitti/postgresql
sudo apt-get update
sudo apt-get install build-essential
sudo apt-get install python-pip
sudo apt-get install python-dev
sudo apt-get install git
sudo apt-get install supervisor
sudo apt-get install rabbitmq-server
sudo update-rc.d rabbitmq-server defaults
sudo apt-get install kannel
sudo apt-get install postgresql-9.2
```

Once the prerequisites are installed, you need to and configurations that will suit the mTrac set-up. A detailed description for such configuration can be accessed on our wiki using the URL below.  
<http://wiki.unicefuganda.org/wiki/ProdEnvironment>

### 3 Virtual Environments

The mTrac set-up is broken down into two instances, the production and test instances. Each of these instances runs under a separate python virtual environment.

Virtual Environments help you to have isolated working copies of Python which allow you to work on a specific project without worry of affecting other projects.

Two Python virtual environments were created for each instance.

- test - for the test instance
- prod - for the production instance

Details on how to setup virtual environments can be accessed on the UNICEF Uganda wiki under the following URL:

[http://wiki.unicefuganda.org/wiki/Orientation#Virtual\\_Environment](http://wiki.unicefuganda.org/wiki/Orientation#Virtual_Environment)

## 4 Dependencies installed Using Pip

Both the test and production instance run on the same set of Python packages.

In this section we show what python packages are required and how to install them into any of your virtual environments using pip.

```
Django==1.3
psycopg2==2.0.13
south
django_nose
django_digest
django-extensions
django-uni-form
djappsettings
django-mptt==0.4.2
django-guardian
django_tablib
django-reversion==1.5.2
django-picklefield
django-celery
virtualenvwrapper
python-virtualenv
python_digest
python-dateutil==1.5
python-memcached
DateUtils
pytz
xlwt
xlrd
openpyxl
uWSGI
numpy
PyYAML
amqplib
httplib2
python-Levenshtein
```

You can add the above packages to a single file (say ‘requirements.txt’) and install them with one command like so;

```
pip install -r requirements.txt
```

## 5 Setup Process

Once all the dependencies are installed, it is time to add the mTrac Django project to the document root of your web server, in this case Nginx. The default is ‘/var/www/’ on Ubuntu.

### 5.1 Clone mTrac git repository

Under the documentroot, add a subdirectories for the production and test instances. These must be readable and writeable by the www user and www group.

```
mkdir -p /var/www/prod
mkdir -p /var/www/test
```

In each of these subdirectories, run the following commands to clone and update the mTrac repository:

```
git clone 'https://github.com/unicefuganda/mtrack.git'
git submodule init
git submodule update
```

### 5.2 Add Logging Directories

When mTrac is running, it actually keeps logs in different log directories/files under the syslog directory, typically ‘/var/log’

Consider running the following commands:

```
sudo mkdir -p /var/log/rapidsms
sudo chown -R www:www /var/log/rapidsms
sudo mkdir -p /var/log/uwsgi
sudo chown -R www:www /var/log/uwsgi
```

### 5.3 Running mTrack

The process of running or stopping is controlled by the supervisor daemon. For more details on setting up and configuring supervisor, please refer to the link shared in chapter 2