Wardrive - README

Sanil Sinai Borkar
{*sanilborkar@ufl.edu*}
Suresh Koppisetty
{*sureshk@ufl.edu*}

December 8, 2014

1 Architecture

2 Setup Instructions for starting a new server

1. Install Nginx:

   ```
   sudo apt-get install nginx
   sudo service start nginx
   ```

   Make sure that you link the ”nginx config” file and ”uwsgi parameters” for the application to nginx. They are given in the code repository.

2. Install Postgres:

   ```
   sudo apt-get install postgresql postgresql-contrib
   ```

3. Install Redis:

   ```
   sudo apt-get install redis-server
   ```

4. Install Django:

   Use this link for installing django the right way http://www.jeffknupp.com/blog/2013/12/18/starting-a-django-16-project-the-right-way/

   Gist of the above article:

   - Install pip
     Check http://pip.readthedocs.org/en/latest/installing.html
   - Install Django and Virtualenv

     ```
     sudo pip install virtualenv
     pip install virtualenvwrapper
     ```

   - Add these entries in .bashrc

     ```
     export WORKON_HOME=~/.Envs
     source /usr/local/bin/virtualenvwrapper.sh
     ```

   - Create virtual env for wardive

     ```
     mkvirtualenv wardrive
     workon wardrive
     ```

5. Project setup

   Clone the repository from the github

   ```
   git clone git@bitbucket.org:uf_sensei/watchtower14.git
   ```

   Once you are done with this basic setup , and run the following command to install all the requirements for the project.

   ```
   pip install -t requirements.txt
   ```

   Make sure that you are already in the virtualenv created previously (”wardrive”) and linking the correct requirements.txt file from the project.

6. Starting the server

   - Start Redis server

     ```
     sudo service start redis-server
     ```

   - Start django using the uwsgi config file.

     ```
     uwsgi --ini wardrive_uwsgi.ini
     ```

   - Start the analysis script as a daemon

     ```
     python Analysis.py daemon
     ```

Django currently supports 3 pages.

- ’/’ - just shows the url for save data. No need in production scenarios
- ’/save_data’ - This url is used to store the data into the postgres server.
- ’/read_data’ - To display analysed data - only for demo purposes, Needs to be disabled in the production environment. This shows all the live data received and analysed. Near real-time.

Changes can be made to the views.py file in the logger folder. Url changes needs to be done in the urls.py file.

3 Android:

1. Import the code to eclipse
2. Dependencies include appcombat_v7. Import it from the android project extras - available with adt extras.
3. Files and their functionality
   - HomeFragment.java - It deals with the fetching logs and displaying it on the screen.
   - MainActivity.java - Starting file which creates 3 fragments - Home , Logs and Credits
   - Method1Fragment.java - Activity file for the Logs tab of the app
   - Method2Fragment.java - Activity file for the Credits tab
   - DBAdapter.java - Addapter to sqlite db of android
   - GatherStats.java - All functions in generating the logs are kept here
   - Stats.java - Stats class file
   - StatsDataSource.java - DB source for sqlite
   - TabsAdapter.java - For getting three tabs - (Home, Logs and Credits)
   - AppController.java - Volley dependency file
   - JSONDataClass.java - Volley file to send logs to server