Instructions to setup and configure 'Plantronixs' web service

Thomas Rhodes

April 12, 2022

CONTENTS

Part I
For Linux Users

PYTHON SETUP

1.1 Check Python

Check to see if you have a compatible version of python installed, either by running `$ python --version` or `$ python3 --version` if successful it should look something similar to the following.

```
[trhod17@k9 ~] $ python --version
Python 3.10.2
```

If if worked move to step 2, if not continue to step 1.2.

1.2 Installing Python

Install python by running the correct command for your distrobution

Debian or Ubuntu based systems:

```
$ sudo apt install python3
```

Arch based systems:

```
$ sudo pacman -S python
```

```
$ sudo pamac install python
```

RHEL based systems:

```
$ sudo dnf install python3
```

Once installed retry step 1.1 to make sure its installed properly

SETTING UP THE ENVIRONMENT

2.1 Getting The Source Code

Go to the Github repo (click here) and Download as a ZIP, then extract to your desired location or clone the repo by using

```
$ git clone https://github.com/Trhod17/tafe_plant_water_timer_app_backend
```

2.2 Setting Up Python Virtual Environment

Now you can either create a new python virtual environment, Guide Here, or use the existing environment saved in the repo by running

For Bash Consoles

```
$ source django_tafe/bin/active
```

For Zsh and other consoles

```
$ . django_tafe/bin/active
```

in the folder containing the source code.

2.2.1 Created New Python Virtual Environment

If you have chosen to create new python virtual environment you next step is to install the required packages by navigating to the directory containing the source code and running
`$ pip install -r requirements.txt`

2.3 Run Django

Once you have finished the above steps run

```
(env) [trhod17@k9 django-tafe]$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced). April 03, 2022 - 08:10:55
Django version 4.0.3, using settings 'plantronics.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

And the website should now be accessible via http://127.0.0.1:8000

2.4 Proj 2 Specifications

Please refer to the other pdf document located in the same directory as this file called RHODES_459274825_Proj2.pdf, on how to test this web service against the specifications for Proj 2, Examples are also given for clarity.

2.5 Proj 2, 18

This step is for when you are up to number 18 in the other document, go to roughly line 174, and change the value of 'burst' rate to 100/second then save, open a new console in the project root directory and or in your IDE, open the python environment as specified above and run `$ pythom manage.py test` and this should run 8 test cases which should each container 3 tests

Part II
For Windows Users

> Disclaimer:
> While the steps below SHOULD work, I do not own a windows machine, nor do i have a virtual machine setup to test these instructions therefore it is of great preference to the author/ developer that this is ran in a Linux environment, as intended. This can be done within windows by installing a Linux distro as an App/program via WSL or by way of a virtual machine

INSTALL PYTHON

Go to www.python.org/downloads/release/python-3014/ and select the appropriate installer for your version of Windows. Download and run the installer, This will install Python on your machine so that we may proceed.

SETUP ENVIRONMENT

Go to the Github repo (click here) and download as a ZIP and extract to desire location, or use git clone to clone the repo and open in desired ide.

CREATE AND SETUP VIRTUAL ENVIRONMENT

5.1 Create Virtual Environment

Now we must create a virtual environment, so activate a terminal in your IDE and type `$ python -m venv env` or `$ python3 -m venv env`, you will see a new folder created called 'env', we need this so don't delete it.

In a terminal within your IDE type the following command: `$ env\bin\activate` to activate the Python Virtual Environment needed in order to setup this web service.

This Python Virtual Environment holds all the packages and dependencies necessary for this web service to run, it works independently of any pre-existing globally installed packages, as a self contained file system.

5.2 Setup Virtual Environment

Now we must install all our package dependencies by typing into the terminal
`$ pip install -r requirements.txt`.

5.3 Launching Django

In the same terminal, with our (env), type `$python manage.py runserver` to start the web service. The web service will be accessible by default via http://127.0.0.1:8000. You should get a similar terminal readout to:

```
(env) $ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 03, 2022 - 08:10:55
Django version 4.0.3, using settings 'plantronics.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

5.4 Proj 2 Specifications

Please refer to the other pdf document located in the same directory as this file called RHODES_459274825_Proj2.pdf, on how to test this web service against the specifications for Proj 2, Examples are also given for clarity.

5.5 Proj 2, 18

This step is for when you are up to number 18 in the other document, go to roughly line 174, and change the value of 'burst' rate to 100/second then save, open a new console in the project root directory and or in your IDE, open the python environment as specified above and run `$ pythom manage.py test` and this should run 8 test cases which should each container 3 tests

END OF DOCUMENT.

THANKS FOR READING!