CMPS 115 SEADS User Documentation

Django Frontend

This is a preliminary setup for our seads Frontend website, which will eventually include d3 visualizations. Currently, this website will only work with a local postgres database, which can be installed with the instructions below.

Installation-Ubuntu 14.04

The django application within this repository must, at this time, work with a local postgres database on your machine. To install postgress on Ubuntu, type the following commands in your terminal:

```
$ sudo apt-get update
$ sudo apt-get install python-pip python-dev libpq-dev postgresql postgresql-contrib
```

Installation-OSX 10.10

This assumes you have the package manager homebrew installed, if not you can go tohttp://brew.sh/ to download and install homebrew.

To install the necessary dependencies of this project open a terminal and type the following commands.

1. Install python3
2. `brew install python3`
3. Install django

4. `pip3 install django`
5. Install requests library for python
6. `pip3 install requests`
7. Install postgresql
8. `brew install postgress`
9. Initialize database
10. `initdb /usr/local/var/postgres`
11. Launch database
12. `postgres -D /usr/local/var/postgres`
13. Or configure it to launch as a background process on login
14.
    ```
    mkdir -p ~/Library/LaunchAgents
    ln -sfv /usr/local/opt/postgresql/*.plist ~/Library/LaunchAgents
    launchctl load ~/Library/LaunchAgents/homebrew.mxcl.postgresql.plist
    ```

Install psycopg2

In order to allow Django to communicate with PostgreSQL you will need to install psycopg2. To do this run:

```
pip3 install psycopg2
```

Note: if pip is having trouble finding you `pg_config` check out this article for help.

Setup local database for project development

Now, log in to postgres as an administrator, and create your database myDB: (Make sure that your postgres is running before attempting this)

Ubuntu 14.04

```
$ sudo su - postgres
```

OSX 10.10+

```
psql -d postgres
```

The rest of these instructions are the same on OSX and Ubuntu

```
$ psql
> CREATE DATABASE myDB;
```

Next, create your username and password(keep the single quotes around 'myPassword'):

```
> CREATE USER myUsername WITH PASSWORD 'myPassword';
```

You should also be sure to explicitly grant all priveleges to your new user:

```
GRANT ALL PRIVILEGES ON DATABASE myDB TO myUsername;
```

Now, quit out of psql:

```
> \q
```

Now, we can configure the Django server in this repository to work with your myDB database. Go into settings.py, and scroll down until you see the object DATABASES. Modify it to look like this:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'myDB',
        'USER': 'myUsername',
        'PASSWORD': 'myPassword',
        'HOST': 'localhost',
        'PORT': '',
    }
}
```

You also need to migrate these changes. Go to the main directory with manage.py, and type:

```
$ python3 manage.py makemigrations
$ python3 manage.py migrate
```

Running the Website

you can now run the existing Seads website! Just type:

```
$ python3 manage.py runserver 0.0.0.0:8000
```

This will run the django server on your machine, at port 8000. Open your browser, and type in 0.0.0.0:8000 at youre address bar. You should see the Seads homepage.

To view the admin page, go to 0.0.0.0:8000/admin. I'm not sure what the proper username and passwords are yet.

To see what the former groups tried to do to visualize data, go to 0.0.0.0/visualization/1. This will take a long time to load.

DB (Backend)

Database Team's repo

Contents

- Deliverables: Scrum project management documentation
- api: Python API database interaction
- puppet: Puppet modules for server configuration after server configuration
- landingzone: Go server to listen for connections to store sensor data
- *Vagrantfile*: Vagrant configuration for development environment
- *deploy.sh*: Deployment shell script to be executed

Using the Vagrant Environment

- Vagrant uses the VirtualBox provider by default, so you will first need to install VirtualBox or find a suitable Vagrant extension for your virtualization platform
- Download and install Vagrant from https://www.vagrantup.com/downloads.html
- From this directory, execute `vagrant up` to create the virtual machine, which will be provisioned after boot
- See the Vagrant docs for more information on using the cli: https://docs.vagrantup.com/v2/cli/index.html

Deplopying to a Server

- On a clean Ubuntu Server 14.04 x64 install, run:

```
sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get install git puppet
git clone https://github.com/seadsystem/Backend.git
cd Backend/DB
sudo puppet apply --modulepath=puppet/modules puppet/manifests/site.pp
./deploy.sh
```

- Optionally, install the latest version of Go with godeb.