
Doc Test Documentation

Release 0.0.1

Dylan Reinhold

Mar 17, 2022

Contents

1	Installation Test	3
1.1	System Prerequisites	3
1.2	Assumptions	3
1.3	Getting Trunk Player	3
1.4	Setup Virtual Environment	4
1.5	Activate Virtual Environment	4
1.6	Install Python Packages	4
1.7	Configure for first use	4
2	Indices and tables	7

Contents:

This document will show you how to get up and running with Trunk Player.
Currently going to be based on a Debian like system Ubuntu

1.1 System Prerequisites

- Linux Machine it might work in windows but has not been tested
- Python 3.5.x
- Virtualenv
- PIP Should be installed with Python 3
- Redis 3.x.x
- git

1.2 Assumptions

- Project directory is `/home/radio/trunk-player`
- Redis is running and listening on default port of `:6379`

1.3 Getting Trunk Player

The source is on GitHub, use git to clone the repository. Starting from your home directory of `/home/radio`

```
$ git clone https://github.com/ScanOC/trunk-player.git
```

This will pull down the most current version of Trunk Player.

1.4 Setup Virtual Environment

Create the directory to run Trunk Player from

```
cd trunk-player
```

Setup a new Python 3.x virtual environment in the `env` directory. Set the visual prompt to `(Trunk Player)`.

```
$ virtualenv -p python3 env --prompt='(Trunk Player)'
```

1.5 Activate Virtual Environment

You will need to re run this step each time you start a new shell or log into your machine.

```
$ source env/bin/activate
```

This will set you into a new python environment any packages you install via `pip` will only live in this area and do not touch your system files. This allowed you to have multiple projects with different dependencies.

You can use the command `deactivate` to exit back to your normal system environment.

1.6 Install Python Packages

Using `pip` install all required packages from the `requirements.txt` file.

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

1.7 Configure for first use

You will need to setup a local version of the `setting.py` file, create and inilize the database, and create a default admin account.

1.7.1 Local settings file

You will need to create a local settings file to override any settings in the `trunk_player/settings.py` file. This will allow you to pull down updates from GitHub without losing your local settings.

Make a copy of the sample local settings file

```
$ cp trunk_player/settings_local.py.sample trunk_player/settings_local.py
```

Important You need to set/change the `SECRET_KEY` in the `trunk_player/settings_local.py`. This value is used to protect sensitive data like passwords. If you keep the one from the project a bad actor may be able to compromise your site or worse your server. See the django project about [SECRET_KEY](#).

1.7.2 Initialize the database

Using the `django manage.py` command to build the new database.

```
$ ./manage.py migrate
```

1.7.3 Create admin account

```
$ ./manage.py c. code-block:: console

$ createsuperuser
Username: test
Email address: test@sample.com
Password: mypassword
Password (again): mypassword
Superuser created successfully.
```

1.7.4 Starting the test web server

First note this is not full production ready server. It can handle a couple users.

Using the `manage.py` command again

```
$ ./manage.py runserver
```

This will start the server up listening on the local loopback address on port 8000. Start your web browser and go to <http://localhost:8000>. You should see the main page Visit `/admin/` to log into the admin area.

If you are running this on a remote server you need to have the web server use its' public IP address so you can connect.

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

This will run the server also on port 8000 but will be accessible via the servers IP address or dns name on port 8000 also.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`