Installation

Intended audience: Site administrator

This tutorial assumes an understanding of common system administration principles and tasks.

Acquiring Theater Ticket Manager (TTM):

Theater Ticket Manager (TTM) is built on the Django framework and utilizes the powerful Docker platform for deployment. It may be deployed through Docker and does not require any other dependencies if so. Once TTM is deployed on a server, it may be accessed by users from any other device by browsing to the server on port 8000 in a web browser.

First, get the latest copy of TTM by running the command below in a terminal:

- git clone https://github.com/aklomkaew/theaterTicketManager.git

Docker
Docker can be used to easily deploy TTM on a webserver. If Docker is used, then Django, Python, and a database server do not need to be installed. The provided Docker image can be used on any platform supported by Docker.

Install Docker:
Docker provides detailed installation instructions for all supported platforms. Please follow their instructions for installation:

- General Docker installation guide: https://docs.docker.com/v17.12/install/
- Mac (easiest option for localhost): https://docs.docker.com/docker-for-mac/install/
- Ubuntu (simplest Linux option): https://docs.docker.com/install/linux/docker-ce-ubuntu/
- CentOS (most flexible option): https://docs.docker.com/install/linux/docker-ce/centos/
- Windows (least support available): https://docs.docker.com/docker-for-windows/install/

Install Docker Compose:
Docker Compose is an official Docker tool that greatly simplifies usage of Docker. Official installation instructions are provided on the Docker website: https://docs.docker.com/compose/install/#install-compose

Local Deployment:

TTM may be deployed through Docker on a local network. It may also be deployed to a DMZ for external use or through a cloud service. This installation guide will only cover local deployment. The steps for installation are essentially the same for local deployment and deployment to any other machine or environment. The primary difference would be in how TTM is exposed to the network, rather than its setup. As for cloud deployment, the process is entirely dependent on the service and is therefore out of scope of this tutorial.

NOTE: Depending on your environment, you may need to run these commands with administrative privileges.

Scripted Management:

Bash scripts are provided for installation, updating, and starting the server. These scripts are provided for ease-of-use. Manual instructions are also provided below.

- To install, run install.sh.
- To update, run update.sh.
- To start the server, run start.sh.

TTM ships with an empty database in order for users to be able to populate it with their own performances and tickets. We have included a pre-populated demonstration dataset that can be used to run the site with sample data. To import this dataset, follow the instructions on the next page.

Manual Installation and Management:

The following steps detail how to manually install TTM. This may be useful if the scripts do not work, or if an explanation is desired for each part of the management process.

Build the Docker Image

Navigate to the directory where the TTM Git repo was cloned to. Enter the 'mySite' directory. You should now be in the working folder for TTM. You should see at least the following files: 'manage.py', 'Dockerfile', 'docker-compose.yml', and 'data_set.json'.

From a command-line prompt, run the following command to prepare the Docker image and install all dependencies:

- `docker-compose build`

Run the following commands in sequence in setup the database:

1. `docker-compose run web python3 manage.py makemigrations`
2. `docker-compose run web python3 manage.py migrate`

Import a Dataset

If you wish to import a dataset, run the command below. Make sure to replace 'data_set.json' with the appropriate filename if you are using a custom dataset.

- `docker-compose run web python3 manage.py loaddata data_set.json`

Run the Docker Image

Once the commands above are run, the Docker image should be prepared. To start the server, run the command below. It should start the server on 0.0.0.0:8000. The server may be accessed through a web browser at localhost:8000, or from another machine at server_ip:8000. Confirm that the firewall allows access to port 8000.

- `docker-compose up`

Test that the server is running by navigating to http://localhost:8000/ in a web browser. The Home page should appear.