

MTurk Framework Deployment Document

By Akshat, Jonathan, Muhammad, Walker

April 11, 2021

Setting Up

To install dependencies

We are going to use a python virtual environment to avoid dependency problems. First, create a virtual environment in the root directory of your local repo.

Then, activate the virtual environment (assuming venv is the name of your virtual environment):

Linux/Mac: `source venv/bin/activate`

Windows: `./venv/Scripts/activate`

While the venv is active, install dependencies from `requirements.txt`:

`pip install -r requirements.txt`

To deactivate the virtual environment, simply run `deactivate`.

Note: ALL required libraries can be found in the `requirements.txt` file.

To set environment variables

Create a `.env` file under `mturksite/mturksite` folder and set the following variables:

- `AWS_ACCESS_KEY_ID`
- `AWS_SECRET_ACCESS_KEY`
- `AWS_REGION_NAME`
- `AWS_ENDPOINT_URL`
- `HOST`

Note: AWS variables are shown when you sign into your AWS account (<https://aws.amazon.com/>) and the HOST variable is your IPV6/IPV4 address

To set user logs

Create a `logs` folder under `mturksite`, as well as a `users.log` file under that same folder. (Should look like this: `mturksite/logs/users.log`)

To run our web app

Make sure to cd into the correct folder of `mturksite/mturkapp`.

When all is set and done, start our web app by running the following command:

`python manage.py runserver`

Remember to set your editor/IDE to use this virtual environment and have it activated when necessary while developing

PostgreSQL

We are going to use PostgreSQL as a backend database as it is well-suited for production use.

To avoid the headache of everyone having to set up a local PostgreSQL server and attempting to keep data synced while developing, we are going to make use of a cloud PostgreSQL server hosted on a Cybera instance. **WARNING: THIS MEANS THAT ANY DATABASE CHANGES WILL AFFECT EVERYONE.**

Connection Information

You will find the connection information for our PostgreSQL server in `mturksite/mturksite/settings.py` in the `DATABASES` dictionary.

Installation

It is strongly recommended that you install pgAdmin 4 (<https://www.pgadmin.org/>) to use as a PostgreSQL management tool. Once installed, make sure you connect to our Cybera hosted PSQL server using the information found in the settings file above.

Remember that you will have to use a VPN tool in order to connect if you do not have IPV6 (see tutorials on eClass).

For Future Developers

Game - OS Installations

To make changes to our existing game, you must first install/download depending on your OS system.

Linux/Mac:

`sudo apt update`

`sudo apt install redis-server`

Further documentation:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-redis-on-ubuntu-18-04>

Windows:

Download Memurai (<https://www.memurai.com/get-memurai>)

Game - Changing Domain Name

Under `mturksite/mturkapp/templates/games`, replace `{domain-name}` with your own in 2 of the files shown below:

1. `question.xml` file on line 3, where `https://{domain-name}/waitPage`.
2. `game.html` file on line 56, where `wss://{domain-name}/ws/gamer`.

More on `channels`, `consumers.py` and `asgi()` programming:

<https://channels.readthedocs.io/en/stable/installation.html>

Amazon MTurk

You're going to need 2 Amazon sandbox accounts for testing your specific applications/tasks.

Remember to access the sandbox accounts, you will first need to have an existing AWS account

Requester Sandbox Account:

<https://requestersandbox.mturk.com/>

Worker Sandbox Account:

<https://workersandbox.mturk.com/>