

Overloop Django Tech Test Backend

Running with Docker

- A sample Dockerfile and a docker-compose is provided to run the application in an isolated environment
- Make sure you have `docker` and `docker-compose` installed and that the Docker daemon is running
- Build and run the container: `docker-compose up`
- Start making some requests: `curl http://localhost:8000/articles/`

Running with a virtual environment

- To run the application in a virtual Python environment, follow these instructions. This example will create a virtual Python environment for 3.9.6
- Check you have the pyenv version you need: `pyenv versions`
- You should see 3.9.6
- If you do not have the correct version of Python, install it like this: `pyenv install 3.9.6`
- On command line do this: `~/.pyenv/versions/3.9.6/bin/python -m venv env`
- This creates a folder called env. Then do this to activate the virtual environment: `source env/bin/activate`
- Lastly do this to check that you are now on the correct Python version: `python --version`
- You can install the dependencies with `pip install -r requirements.txt`
- You should run `python setup_and_seed.py` to get a local database setup and seeded with lookup data
- You can then run the app with `python manage.py runserver 0.0.0.0:8000` in the root directory

Project Structure Notes

- There are two django apps installed `articles` and `regions`
- Django is used as a RESTful API, no html rendering is required
- Marshmallow is used to serialize and deserialize django object instances

Tasks

- Add an new entity called `Author` with a `first_name` and a `last_name`. An API user should be able to create a new `Author`, edit an existing one and list all existing ones.
- Update the `Article` entity so that it relates to an `Author`. An API user should be able to select an `Author` and/or `Region` when creating or editing an `Article`.
- A user should be allowed to enter an `Article` with no `Author`, or remove the `Author` from an existing `Article`.
- An API user should be able to perform the following actions for each `Article`, `Author` and `Region` entity:
 - Get all entities
 - Create a single entity
 - Get a single entity
 - Update a single entity

- Delete a single entity
- The app should be robust and you should make sure that everything works as specified.
- Add unit tests for any code written to implement the tasks using a testing framework of your choice.