# Deploying to Heroku

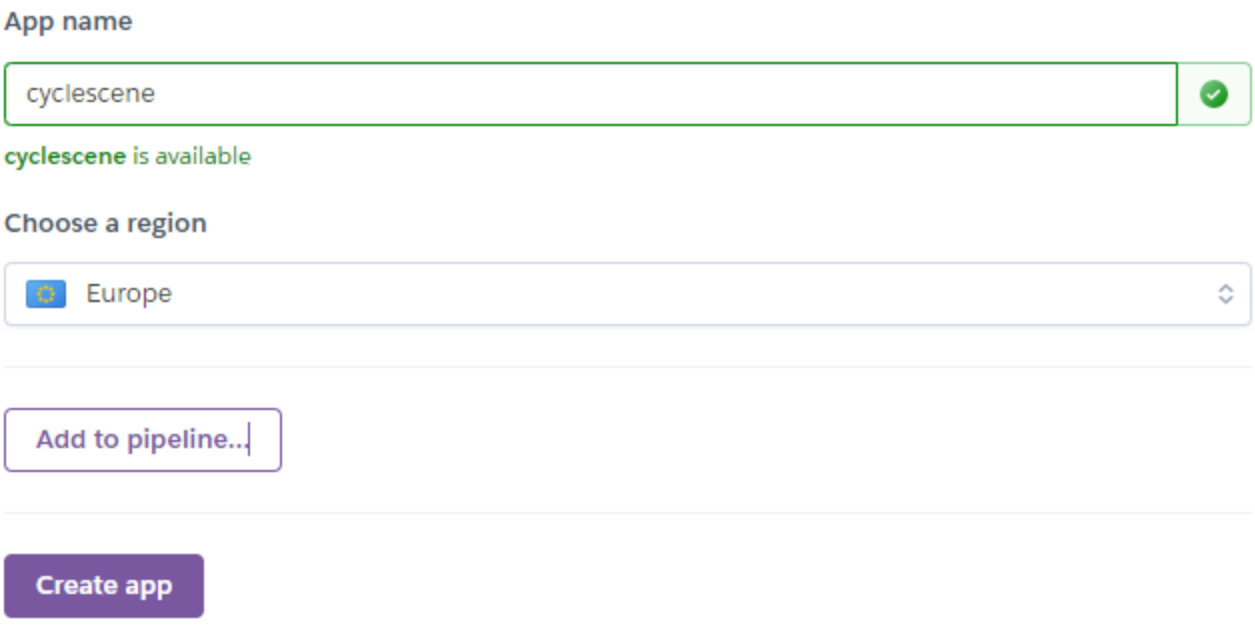
## Prerequisites

We did play around with using fixtures at the start of the project but chose not to use them for our own data since it was a lot faster to use the product manager to enter them individually that than writing up a large json file.

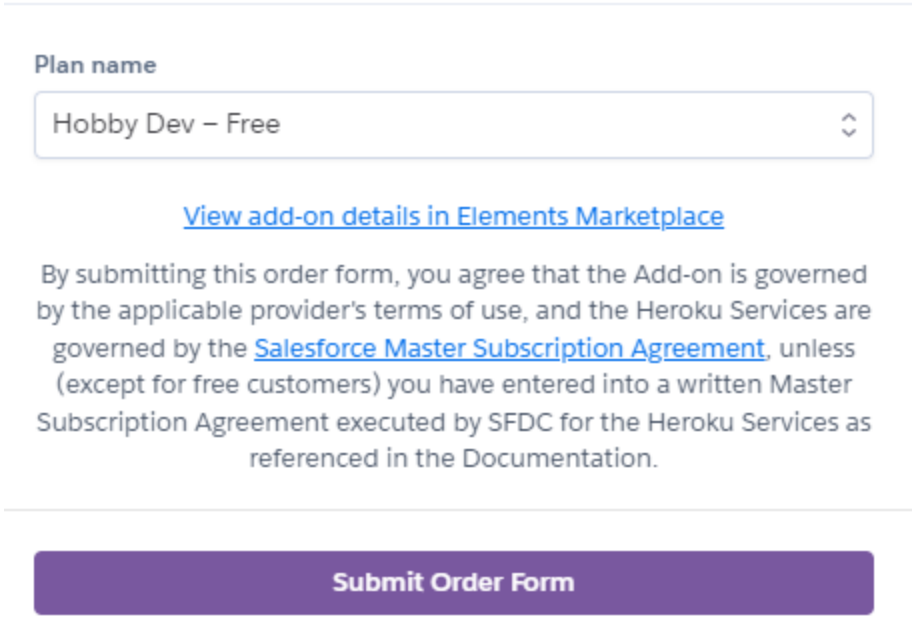
So as per guidance we export our Sql lite DB since we'll be changing over to postgres when the project is finally deployed. To do this we follow the steps below.

1. Used the command **`python3 manage.py dumpdata --exclude auth.permission --exclude contenttypes > db.json.`**
2. Now that we have our json dump file, to restore it into postgress use the command **`python3 manage.py loaddata db.json`**
3. `This creates a db.json file on the root of our gitpod folder.`

1. Now we open need to create the Heroku app, navigate to https://www.heroku.com/ and log into the dashboard.
2. Select the 'new' button in the right hand corner and choose 'create new app'
3. Give your new app a unique name and select the appropriate region, then choose 'create app'.



4. On the 'Resources' tab we need to provision a new database for the project. Here we choose 'Heroku PostgreSQL' and choose the free plan since its only for a project and not a live site.




5. Now to use postgres we head back over to 'GitPod' and install 'Dj Database' and 'Psycopg2' using the following commands.

- pip3 install dj_database_url
- pip3 install psycopg2-binary

and finally update our requirements file using the command below.

- Pip3 freeze > requirements.txt

6. Next we go to the top of our project settings file and import dj_database_url

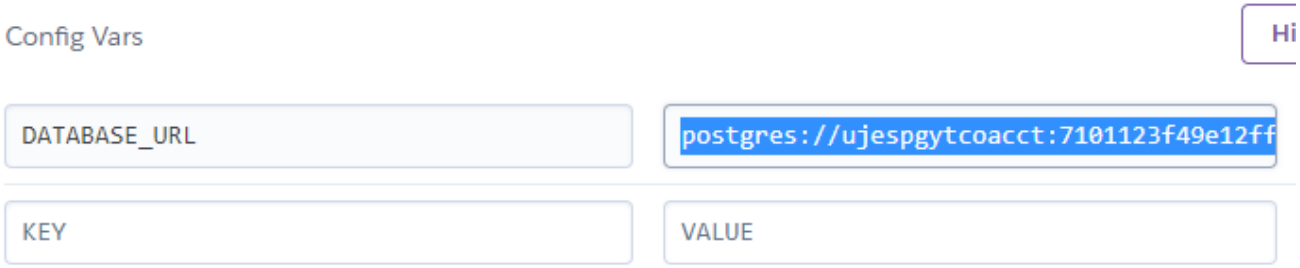
```
import os
import dj_database_url
from pathlib import Path
```

Then we can comment out our default sqlite3 database since we won't be using it, and add in our default database to point to dj_database_url.

```
#   DATABASES = {
#       'default': {
#           'ENGINE': 'django.db.backends.sqlite3',
#           'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
#       }
#   }

DATABASES = {
    'default': dj_database_url.parse('postgres://ujespgytcoacct:7101123f49e12ff818643b0
}
```

7. Now copy in the config vars from the setting in your new Heroku app.




Running:

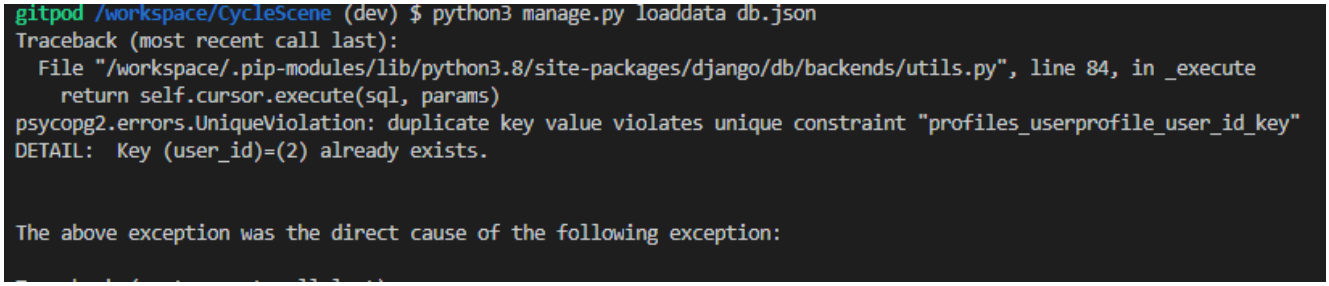
- python3 manage.py showmigrations
- python3 manage.py migrate

This reveals all app migrations need to run so the project can configure itself with the new postgress db.
Now we run manage.py migrate to apply all migrations.

```
 [ ] 0001_initial
shoppingbag
 (no migrations)
sites
 [ ] 0001_initial
 [ ] 0002_alter_domain_unique
socialaccount
 [ ] 0001_initial
 [ ] 0002_token_max_lengths
 [ ] 0003_extra_data_default_dict
 [ ] 0004_auto_20220221_1455
gitpod /workspace/CycleScene (dev) $ python3 manage.py showmigrations
```

After migrations have run we now can reimport our. json file.

I got the following error when importing my database.




After reading up on the topic it turns out there is a user account with an 'id of 1' in my .json dump file which is conflicting with the new postgres db. that trying to create a user account also with the 'id of 1'.

To fix it I did the following steps:

A. I changed back to the local sqllite3 db, deleted all my users since I shouldn't need them anyway and ran the export again so basically without any user profiles.
B. I then changed back to postgres and ran the import and got a success message this time.
C. So as not to leave the old db in an error state. I changed back to sqllite3 db and ran the command **python3 manage.py createsuperuser** to allow me to login as normal if I need to switch back for some reason.
D. Finally, I changed back to postgres and ran the command **python3 manage.py createsuperuser**
E. The site loaded as normal and I could login using the same /admin after the address.

8. Next we need to install gunicorn which will act as our server and then freeze the requirements:

    - Pip3 install gunicorn
    - pip3 freeze > requirements.txt

9. Now that the above steps are completed we can create our Procfile in the root and specify it point to our project.

    - web: gunicorn cyclescene.wsgi:application

10. Next we need to login to Heroku, in my case I got a warning about '*bad ip address*' this was resolved by logging into the cli (heroku login -i). Next we run the command below an importantly at this stage we disable the ability to collect static files since we don't require them yet.

    - heroku config:set DISABLE_COLLECTSTATIC=1 --app cyclescene

```
gitpod /workspace/CycleScene (dev) $ heroku config:set DISABLE_COLLECTSTATIC=1 --app cyclescene
 ›   Warning: heroku update available from 7.59.2 to 7.60.2.
Setting DISABLE_COLLECTSTATIC and restarting ⬢ cyclescene... done, v5
DISABLE_COLLECTSTATIC: 1
gitpod /workspace/CycleScene (dev) $
```

11. We can now add our app to the settings.py file and our new postgres database address as well.

```
ALLOWED_HOSTS = ['cyclescene.herokuapp.com', 'localhost']
```

```
DATABASES = {
    'default':
dj_database_url.parse('postgres://ujespgytcoacct:7101123f49e12ff818643b04f222ed
6d5e0300b217c636630b4dbe4985a81078@ec2-63-35-156-160.eu-west-
1.compute.amazonaws.com:5432/d5oquaqgr76nnm')
}
```

12. We were having an issue with 'internal server error' when trying to setup automatic deploy in Heroku.
    Could not setup automatic deployment because of the recent security breach in Heroku which has removed the ability for automatic deploys.
    To get around this we followed the steps below

    Tried this

    1. Command: **heroku apps**
    2. Command: **heroku git:remote –a cyclescene**

Now we can login to Heroku.

1. Login to heroku and enter your details.
command: **heroku login –i**

2. Get your app name from heroku.
command: **heroku apps**

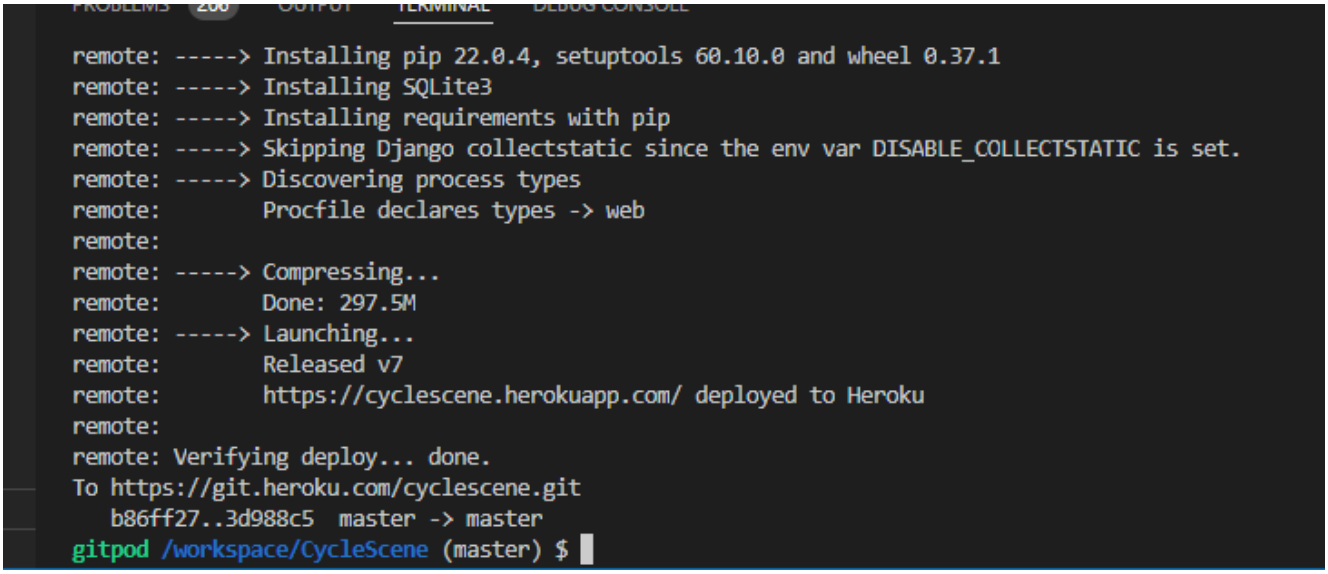
3. command: **heroku git:remote -a cyclescene**

```
gitpod /workspace/CycleScene (master) $ heroku apps
 ›   Warning: heroku update available from 7.59.2 to 7.60.2.
=== heroku@kryan23.bulc.club Apps
cyclescene (eu)
```

4. Add, commit and push to github
command: **git add . && git commit -m "Commit Message"**

5. Push to both github and heroku
command: **git push origin master**
command: **git push heroku master**

6. In this case we updated the config vars with a new secret key and pushed a new build to Heroku.


It was noted that these steps will need to be performed each time we make changes in our development environment to push a new release to Heroku.

13. Scroll up to the top of the screen and choose the 'Open App' button to launch the new app and see the result in the browser.

Our unique Url for the project: https://cyclescene.herokuapp.com/

This finishes the main Heroku app deployment.

end