



# Django-Heroku Upload

Developed by Alabian Solutions Ltd

---

# Django Heroku Upload

## Create a heroku account and download the Heroku CLI

1. Signup to Heroku by visiting this link  
<https://signup.heroku.com/>
2. Download the Heroku CLI for windows by visiting this link  
<https://devcenter.heroku.com/articles/heroku-cli>

## Installing pipenv and creating virtual environment

1. Create a folder on your desktop or anywhere and give it a name
2. Install pipenv by issuing the command below

```
pip install pipenv
```

3. Create a virtual environment by issuing the command below

```
virtualenv .
```

4. Activate the virtual by issueing the command below

```
.\Scripts\activate
```

## Installing django in the virtual environment created

1. On the folder created on your desktop install django by issuing the command below

```
pip install Django==1.10
```

2. Copy the project you are working on into the folder created on the desktop
3. Change directory into your project folder and start installing the packages you have previously eg your tinymce, fontawesome etc. This step is optional if you don't have packages installed in your project.

```
C:\Users\Benedict\Desktop\publish>cd upload_project/
```

4. Run python manage.py runserver to confirm that there is no error in your code.

## Install gunicorn

1. Install gunicorn by issuing the command the below

```
pip install gunicorn
```

## Creating a Procfile and adding content to it

1. Create a procfile and make sure it has no extension, the procfile should be of the format below

```
Procfile
```

2. Add the following content into the Procfile

```
web: gunicorn project_name.wsgi
```

the “project\_name” above will be whatever name you gave project folder

## Installing Django-heroku

1. Install Django-heroku with the command below

```
pipenv install django-heroku
```

2. Add the following at the settings.py file
  - a. Add this at the top of settings.py directly above ‘os’

```
import django_heroku
```

- b. Add this at the bottom of settings.py this can be the last thing on settings.py

```
django_heroku.settings(locals())
```

3. Create a “requirements.txt” file by issuing the command below

```
pip freeze > requirements.txt
```

4. Create a runtime.txt file and add the below code to the file

```
python-3.7.2
```

## Create an your project name and push to git

1. Use the command below to create your project name, make sure the project name you choose is unique, this returns the url of project

```
heroku create yourprojectname
```

2. Initialize your project as a repository and save changes to the repo

```
git init
```

```
git add .
```

```
git commit -m "your commit message"
```

3. Add your project to the remote repo and push

```
heroku git:remote -a yourprojectname
```

```
git push heroku master
```

## How to fix admin not working when trying to login

1. You need to run a bash command to access files that you have already uploaded on the heroku server

```
heroku run bash
```

In the bash make migrations by running the command below

```
python manage.py migrate
```

## Create a super user to login to your admin on your heroku server

1. Use the command below to create a super user on the heroku bash

```
python manage.py createsuperuser
```

2. Use the command below to exit the heroku bash

```
exit
```

Note: if you make any change to your python or html files save changes to repository and use the git push command to the heroku server