

Django Blog Cheat Sheet (full):

Setting up basic Django Project and Deploying to Heroku

Warning: Terminal command update

Since this video was created, Django has introduced a new version that will be automatically installed if you use the command in the video.

To ensure that you get the same version of django and gunicorn used in this video and so that nothing breaks as you do the walkthrough, **instead of the command `pip3 install django gunicorn`, please use this:**

`pip3 install 'django<4' gunicorn`

This will install Django 3.2 which is the LTS (Long Term Support) version of Django and is therefore preferable to use over the newest beta Django 4.

Step 1: Installing Django and supporting libraries

Note: It is recommended when you are still learning this content that you type out each line of code, rather than copying and pasting. This will help you learn!

Key:

PROJ_NAME = **codestar2021**

APP_NAME = **blog**

In the Terminal:

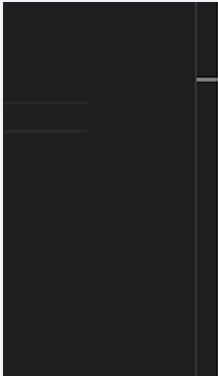
#	Step	Code
1.	Install Django and gunicorn:	<code>pip3 install 'django<4' gunicorn</code>
2.	Install supporting libraries:	<code>pip3 install dj_database_url==0.5.0 psycopg2</code>
3.	Install Cloudinary Libraries	<code>pip3 install dj3-cloudinary-storage</code> <code>pip3 install urllib3==1.26.15</code>

4.	Create requirements file	pip3 freeze --local > requirements.txt
5.	Create Project (codestar 2021)	django-admin startproject PROJ_NAME . (Don't forget the .)
6.	Create App (blog)	python3 manage.py startapp APP_NAME

settings.py

#	Step	Code
7.	Add to installed apps	INSTALLED_APPS = [... ' APP_NAME ',]
*	Save file	

In the Terminal:

#	Step	Code
8.	Migrate Changes	python3 manage.py migrate
9.	Run Server to Test	python3 manage.py runserver
10.	<p>You will see a yellow error screen, don't worry! Your server is running properly. This error is telling you that, for security reasons, Django doesn't recognise the hostname - the server name your project is running on.</p> <p>Select and copy the hostname after "Invalid HTTP_HOST header". In this example, that is '8000-nielmc-django-project-0kylrta3cs.us2.codeanyapp.com' - you can include the quotes.</p>	 <p>Paste the hostname between the square brackets of ALLOWED_HOSTS. For the above example, this would look like</p> <pre>ALLOWED_HOSTS = ['8000-nielmc-django-project-0kylrta3cs.us2.codeanyapp.com']</pre>

--	--	--

Step 2: Deploying an app to Heroku

5 stages:

1. Create a new external database
2. Create the Heroku app
3. Attach the database
4. Prepare our environment and settings.py file
5. Get our static and media files stored on Cloudinary.

Note: Error fix

If you get the error below during the steps to deployment:

`django.db.utils.OperationalError: FATAL: role "somerandomletters" does not exist`

Please run the following command in the terminal to fix it:

`unset PGHOSTADDR`

2.1 Create a new external database

On elephantsql.com:

#	Step	Code	Your Notes
1	Log in to your ElephantSQL account		If you don't have an ElephantSQL.com account yet, the steps to create one are here .
2	Click “ Create New Instance ”		

3	Set up your plan	<ul style="list-style-type: none"> • Give your plan a Name (this is commonly the name of the project) • Select the Tiny Turtle (Free) plan • You can leave the Tags field blank 	
4	Click “ Select Region ”	Select a data center near you	Note: If you receive a message saying "Error: No cluster available in your-chosen-data-center yet", choose another region
5	Click “ Review ”	Check that your details are correct. Then click “ Create instance ”	
6	Return to the ElephantSQL dashboard and click on the database instance name for this project		
7	Copy your ElephantSQL database URL using the Copy		

	icon. It will start with postgres://		
--	--------------------------------------	--	--

2. Create the Heroku app

In heroku.com: (Note: must be logged in)

	Step	Code	
1.	Create new Heroku App	APP_NAME , Location = Europe	
2	Open the settings tab		
3	Click Reveal Config Vars		
4	Add a Config Var called DATABASE_URL		Note: The value should be the ElephantSQL database url you copied in the previous step

2.2 Attach the Database:

In the IDE file explorer or terminal:

#	Step	Code
4.	Create new env.py file on top level directory	E.g. env.py

In env.py

#	Step	Code
5.	Import os library	import os
6.	Set environment variables	os.environ["DATABASE_URL"] = " Paste in ElephantSQL database URL "

7.	Add in secret key	os.environ["SECRET_KEY"] = " Make up your own randomSecretKey "
----	-------------------	--

In heroku.com

#	Step	Code
8.	Add Secret Key to Config Vars	SECRET_KEY, " randomSecretKey "

2.3 Prepare our environment and settings.py file:

In settings.py

#	Step	Code
9.	Reference env.py (Note: font in bold is new)	from pathlib import Path import os import dj_database_url if os.path.isfile("env.py"): import env
10.	Remove the insecure secret key and replace - <i>links to the SECRET_KEY variable on Heroku</i> (Note: font in bold is new)	SECRET_KEY = os.environ.get('SECRET_KEY')
11.	Comment out the old DataBases Section	# DATABASES = { # 'default': { # 'ENGINE': 'django.db.backends.sqlite3', # 'NAME': BASE_DIR / 'db.sqlite3', # } # }
12	Add new DATABASES Section <i>- links to the DATATBASE_URL variable on Heroku</i>	DATABASES = { 'default': dj_database_url.parse(os.environ.get("DATABASE_URL")) }

In the Terminal

#	Step	Code
12.	Save all files and Make Migrations	<code>python3 manage.py migrate</code>

2.4 Get our static and media files stored on Cloudinary:

In [Cloudinary.com](https://cloudinary.com): (Note: must be logged in)

#	Step	Code
1.	Copy your CLOUDINARY_URL e.g. API Environment Variable.	From Cloudinary Dashboard

In env.py:

#	Step	Code
2.	Add Cloudinary URL to env.py - <i>be sure to paste in the correct section of the link</i>	<code>os.environ["CLOUDINARY_URL"] = "cloudinary://*****"</code>

In Heroku:

#	Step	Code
3.	Add Cloudinary URL to Heroku Config Vars - <i>be sure to paste in the correct section of the link</i>	Add to Settings tab in Config Vars e.g. CLOUDINARY_URL, <code>cloudinary://*****</code>
4.	Add DISABLE_COLLECTSTATIC to Heroku Config Vars (temporary step for the moment, will be removed before deployment)	e.g. DISABLE_COLLECTSTATIC, 1

In settings.py:

#	Step	Code
5.	Add Cloudinary Libraries to installed apps	<pre>INSTALLED_APPS = [..., 'cloudinary_storage', 'django.contrib.staticfiles', 'cloudinary', ...,]</pre> <p>(note: order is important)</p>
6.	Tell Django to use Cloudinary to store media and static files <i>Place under the Static files Note</i>	<pre>STATIC_URL = '/static/' STATICFILES_STORAGE = 'cloudinary_storage.storage.StaticHashedCloudinaryStorage' STATICFILES_DIRS = [os.path.join(BASE_DIR, 'static'),] STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles') MEDIA_URL = '/media/' DEFAULT_FILE_STORAGE = 'cloudinary_storage.storage.MediaCloudinaryStorage'</pre>
7.	Link file to the templates directory in Heroku <i>Place under the BASE_DIR line</i>	<pre>TEMPLATES_DIR = os.path.join(BASE_DIR, 'templates')</pre>
8.	Change the templates directory to TEMPLATES_DIR <i>Place within the TEMPLATES array</i>	<pre>TEMPLATES = [{ ..., 'DIRS': [TEMPLATES_DIR], ..., }, },],</pre>

9.	Add Heroku Hostname to ALLOWED_HOSTS (e.g. codestar2021)	ALLOWED_HOSTS = [" PROJ_NAME .herokuapp.com", " YOUR_HOSTNAME "]
----	--	--

In the IDE file explorer or terminal:

#	Step	Code
10.	Create 3 new folders on top level directory	media, static, templates
11.	Create a Procfile on the top level directory	Procfile

In Procfile

#	Step	Code
12.	Add code	web: gunicorn PROJ_NAME .wsgi

*** Note:** Save all files

In the Terminal:

#	Step	Code
13.	Add, Commit and Push	git add . git commit -m "Deployment Commit" git push

In Heroku:

#	Step	Code
14.	Deploy Content manually through heroku/	E.g Github as deployment method, on main branch
