

Table of Contents:

- A. SETUP DJANGO USING DOCKER ON LOCAL COMPUTER
- B. RESUMING WORK ON DJANGO USING DOCKER ON LOCAL COMPUTER
- C. MAKING A HELLO WORLD APP
- D. DISPLAY A PREWRITTEN HTML PAGE (TEMPLATE) IN AN APP
- E. DISPLAY A PREWRITTEN HTML PAGE (TEMPLATE) WITH VARIABLES IN THE PAGE
- F. CREATE A DATABASE AND TABLE IN POSTGRES USING DOCKER
- G. RESUMING WORK ON DATABASE IN POSTGRES USING DOCKER

A. SETUP DJANGO USING DOCKER ON Local Computer

Start docker

Create a folder for your website on your computer called projects

Open VS Code and Open the projects folder

Open Terminal for typing

Type "docker network create donut-net" (this only needs to be done once on a computer with docker)

MAC INSTRUCTION

Type:

```
docker run --rm --net donut-net -p 8080:8080 -v "`pwd`: "`pwd`" -w "`pwd`" -ti python bash
```

PC INSTRUCTION

Type:

```
docker run --rm --net donut-net -p 8080:8080 -v ${PWD}:/usr/local/myprogram -w /usr/local/myprogram -ti python bash
```

Type: pip install virtualenv

Type: python -m venv ./venv

Type: ". venv/bin/activate" #notice the space after the .

Type: pip install Django

Type: django-admin startproject donutwebsite

Type: cd donutwebsite

Type: python manage.py migrate

Type: python manage.py createsuperuser

Change file donutwebsite/donutwebsite/settings.py on line 28 to allow all hosts ['*'] (save file)

Type: python manage.py runserver 0.0.0.0:8080

In a web browser, open <http://localhost:8080/>

Stopping work:

Type Ctl-C to stop server

Type deactivate

Type exit

Quit docker

B. RESUMING WORK ON DJANGO USING DOCKER ON LOCAL COMPUTER

Start docker

Open VS Code and Open the folder for your website

Open Terminal for typing

MAC INSTRUCTION

Type: `docker run --rm --net donut-net -p 8080:8080 -v "`pwd`: "`pwd`" -w "`pwd`" -ti python bash`

PC INSTRUCTION

Type: `docker run --rm --net donut-net -p 8080:8080 -v ${PWD}:/usr/local/myprogram -w /usr/local/myprogram -ti python bash`

Type: `". venv/bin/activate"` #notice the space after the .

Type: `cd donutwebsite`

Type: `python manage.py runserver 0.0.0.0:8080`

In a web browser, open <http://localhost:8080/>

Stopping work:

Type Ctl-C to stop server

Type deactivate

Type exit

Quit docker

C. MAKING A HELLO WORLD APP

1 Open an existing project or create a new Django project

2 In the terminal cd so that your cursor is in the project folder (e.g. cd donutwebsite)

3 Create an app. For aboutus, the syntax is:

```
python manage.py startapp aboutus
```

4 Register the new app in the settings file for your project under `INSTALLED_APPS`. If your project is named donutwebsite, then the settings file is donutwebsite/donutwebsite/settings.py. An app called aboutus gets registered with the name: `'aboutus.apps.AboutusConfig'`

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'aboutus.apps.AboutusConfig',  
]
```

5 In the views.py file for your new app, change the content to be:

```
from django.http import HttpResponse

def indexPageView(request) :
    return HttpResponse('About us Website!')
```

6 In the folder for your new app, create a file urls.py and supply it with the following text.

```
from django.urls import path
from .views import indexPageView

urlpatterns = [
    path("", indexPageView, name="index")
]
```

7 Change the project's urls.py file (e.g. donutwebsite/donutwebsite/urls.py) so it can see your app's urls.py file.

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path("",include('aboutus.urls'))
]
```

8 Start your website

Type: python manage.py runserver 0.0.0.0:8080

D. DISPLAY A PREWRITTEN HTML PAGE (TEMPLATE) IN AN APP

- 1 Create or open an existing Django project
- 2 Create an app within a Django project. E.g. aboutus
- 3 Create a folder called 'templates' in your main project folder. (Not in an app folder)
- 4 Create a subfolder in templates with the same name as your app, e.g. aboutus.
- 5 In the settings.py file for your project, around line 13, add an import statement for the os library. Lines 13 and 14 shown here:

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

6 In the settings.py file for your project, modify the TEMPLATES section so that the project can see your new templates folder. It is the one line with the entry for DIRS.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

6 In the templates/aboutus folder, add a file such as form.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>My Form</title>
</head>
<body>
    <h3>Tell us about you</h3>
    <form>
        <label>Full Name:</label><input type="text" name="FullName"><br>
        <select name="favoritedonut">
            <option>Apple Fritter</option>
            <option>Glazed</option>
            <option>Jelly Filled</option>
        </select>
        <input type="Submit" value="Submit Form">
    </form>
</body>
</html>
```

7 In your app's (aboutus) views.py files, add a library import for render, and a "view" function that will return your form page to an html browser request.

```
from django.http import HttpResponse
```

```
from django.shortcuts import render
```

```
def indexPageView(request) :  
    return HttpResponse('About us Website!')
```

```
def formPageView(request) :  
    return render(request, "aboutus/form.html")
```

8 In your apps urls.py file, add an import statement for your view function at the top and add a route to your new view function.

```
from django.urls import path  
from .views import indexPageView, formPageView
```

```
urlpatterns = [  
    path("", indexPageView, name="index"),  
    path("form", formPageView, name="form")  
]
```

9 Start your website (if not already started)

Type: `python manage.py runserver 0.0.0.0:8080`

10 Visit the url using the path to your new web page:

e.g.

<http://9eq999af0ads0ff8a0f.amazon.com/form> or <http://localhost:8080/form> (second example is running locally)

E. DISPLAY A PREWRITTEN HTML PAGE (TEMPLATE) WITH VARIABLES IN THE PAGE

1 Complete each of the steps for displaying a prewritten HTML page (Template) in an app.

2 In the templates folder, in the app folder (e.g. aboutus) create a new html page, let's do locations.html

Notice that it has two variables, sPagetitle (a string) and sLocation (a list)

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <title>{{sPagetitle}}</title>
```

```

</head>
<body>
  <h2>Page title: {{sPagetitle}}</h2>

  List of locations: <br>
  {% for sLocation in lLocations %}
    {{ sLocation }} <br>
  {% endfor %}

</body>
</html>

```

3 Add a page view function for locations.html within the views.py file for your app. Include a dictionary object with the variables names you plan to use in your template file.

```

from django.http import HttpResponseRedirect
from django.shortcuts import render

def indexPageView(request) :
    return HttpResponseRedirect('About us Website!')

def formPageView(request) :
    return render(request, "aboutus/form.html")

def locationsPageView(request) :
    # some variables in a dictionary object to pass into the template page
    dContext = {
        "sPagetitle": "Locations Page",
        "lLocations": ["Provo", "Ogden", "Moab", "Paris"]
    }
    return render(request, "aboutus/locations.html", dContext)

```

4 In the urls.py file, import the new page view definition and provide a path to the page view function.

```

from django.urls import path
from .views import indexPageView, formPageView, locationsPageView

urlpatterns = [
    path("", indexPageView, name="index"),
    path("form", formPageView, name="form"),
    path("locations", locationsPageView, name="locations")
]

```

5 10 Visit the url using the path to your new web page:

e.g.

<http://9eq999af0ads0ff8a0f.amazon.com/locations> or <http://localhost:8080/locations> (second example is running locally)

F. CREATE A DATABASE AND TABLE IN POSTGRES USING DOCKER

1 In your terminal, change directory to your projects folder (e.g. 'projects')

2 Create folder to store your database files, call it db.

3 Run the following command at the terminal (all on one line!!!!):

MAC INSTRUCTION

```
docker run --name donutdb --net donut-net -e POSTGRES_PASSWORD=Password1 -d -v  
`pwd`/db:/var/lib/postgresql/data postgres
```

PC INSTRUCTION

```
docker run --name donutdb --net donut-net -e POSTGRES_PASSWORD=Password1 -d -v  
${PWD}/db:/var/lib/postgresql/data postgres
```

A database is now running in the background.

4 Connect your terminal to the database.

Type: `docker exec -ti donutdb bash`

5 Log into a postgres in interactive mode

Type: `psql -U postgres`

6 Create a database called paradisedonutdb

Type: `create database paradisedonutdb;`

7 Change the current working database to paradisedonutdb

Type: `\c paradisedonutdb`

8 Create an employee table.

Type:

```
CREATE TABLE employee
(
    emp_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 ),
    first_name varchar(30) NOT NULL,
    last_name varchar(30) NOT NULL
);
```

9 Add a record to the employee table.

Type:

```
INSERT INTO employee (first_name, last_name) VALUES ('Jane','Vasquez');
```

10 View all records in employee table

Type:

```
SELECT * FROM employee;
```

11 Exit database and container:

To exit psql type: \q

To exit container with postgres running type: exit

G. RESUMING WORK ON DATABASE IN POSTGRES USING DOCKER

1 Connect to your running database: (if docker is still running since the last time you used the computer), in a terminal, type*:

```
docker exec -ti donutdb bash
```

2 Log into a postgres in interactive mode

Type: psql -U postgres

3 Change the current working database to paradisedonutdb

Type: \c paradisedonutdb

4 View all records in employee table

Type:

```
SELECT * FROM employee;
```


*IF YOUR DOCKER PROGRAM IS STOPPED (e.g. you rebooted your computer recently), you will get an error message that says "Cannot connect to Docker daemon".

-Start Docker on your computer.

-In the terminal, type: `docker start donutdb`