

## How to create a docker image :

Docker gives you the capability to create your own Docker images, and it can be done with the help of Docker Files. A Docker File is a simple text file with instructions on how to build your images.

Create a file called Docker File and edit that by writing appropriate commands to build your image. Please note that the name of the file has to be "Dockerfile" with "D" as capital.

Ex: Creating a docker image for a django application

1. Create a requirements.txt file and write all dependencies and packages needed to install to create image of that django application .

```
--->requirements.txt
#For this application requirements are
django
django-allauth
django-extensions
django-model-utils
django-rest-auth
django-webpack-loader
djanoorestframework
djanoorestframework-jwt
```

2. Create a Dockerfile in the root directory of your web application and in that Dockerfile write the appropriate commands to build a container running a Django web application server. The vast majority of Dockerfiles will begin by referencing a base image provided by Docker. Take “django:python3-onbuild” as base image .

(you can also take python as base image) .And then set up directory structures, environment variables, download dependencies, and many other standard tasks before finally executing the process which will run your web application.

```
# Dockerfile
# FROM directive instructing base image to build upon
FROM django:python3-onbuild
```

3. Copy all the files which are responsible for web application into image.

#COPY - copies the specified files into image (container) .  
#here djangoapp copied is the full folder which contains djangoapplication and requirements.txt

COPY djangoapp djangoapp

4. Change the working directory to djangoapp where it has requirements.txt and manage.py

#WORKDIR - change the working directory  
WORKDIR djangoapp

5. Now install all the packages in requirements.txt

#RUN - runs the command given.  
RUN pip install requirements.txt

6. Give a port to run the application

EXPOSE 9123

7. Give commands to run your django application.

# CMD-is the command the container executes by default when you launch the image.  
CMD ["python", "manage.py", "migrate"]  
CMD ["python", "manage.py", "runserver", "0.0.0.0:9123"]

8. Now the Dockerfile looks like

```
#Dockerfile
FROM django:python3-onbuild
COPY djangoapp djangoapp
WORKDIR djangoapp
RUN pip install requirements.txt
EXPOSE 9123
CMD ["python", "manage.py", "migrate"]
CMD ["python", "manage.py", "runserver", "0.0.0.0:9123"]
```

9. Now go to directory where Dockerfile is present and run the below command.

```
$ docker build -t <imagename:tag> .
```

(At last the dot '.' is important to mention that the Dockerfile is present in working directory)

10. Now the image was built .Run the command “docker images” , you will see your image.

11. Now run the image as container.

```
$ docker run -p 9123:9123 <imagename:tag>
```

Check localhost:9123 in your browser, which shows your application.