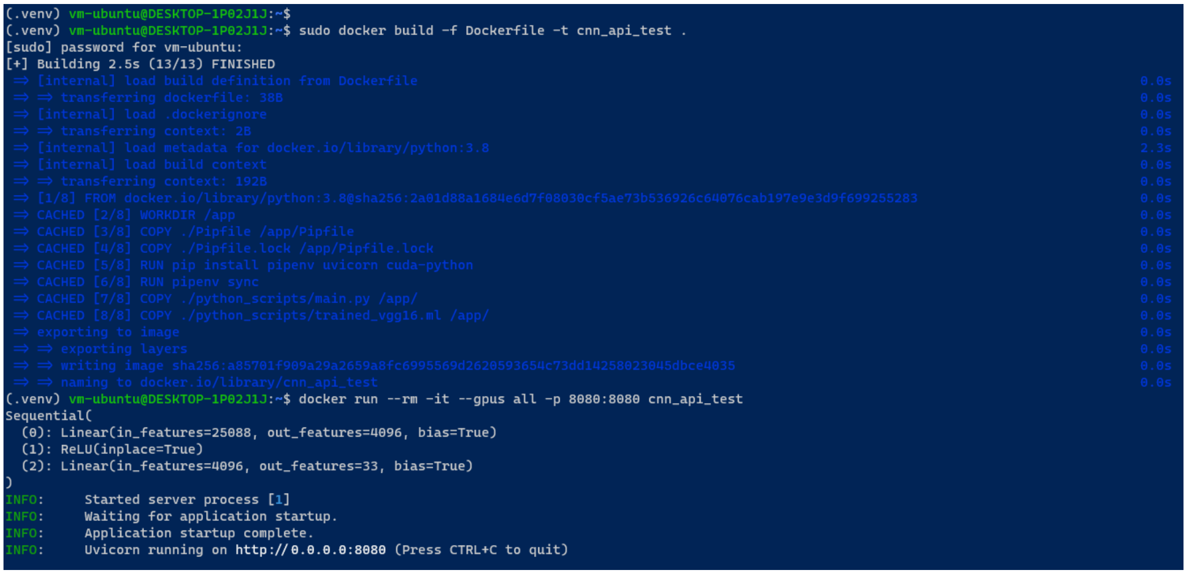
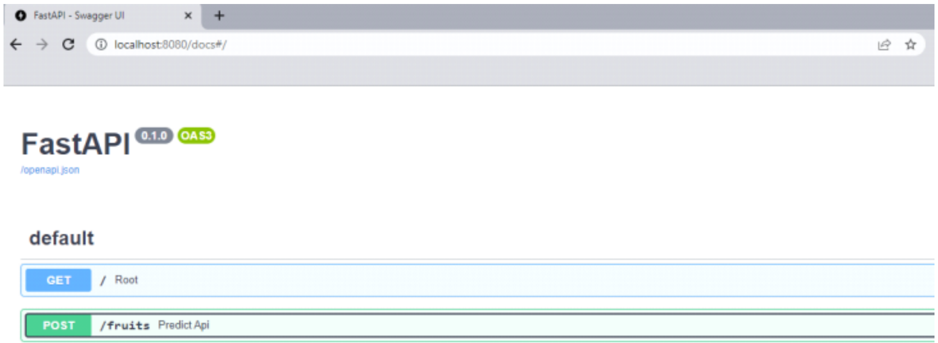
1. Build docker image from Dockerfile. Create and run a contained from this image.

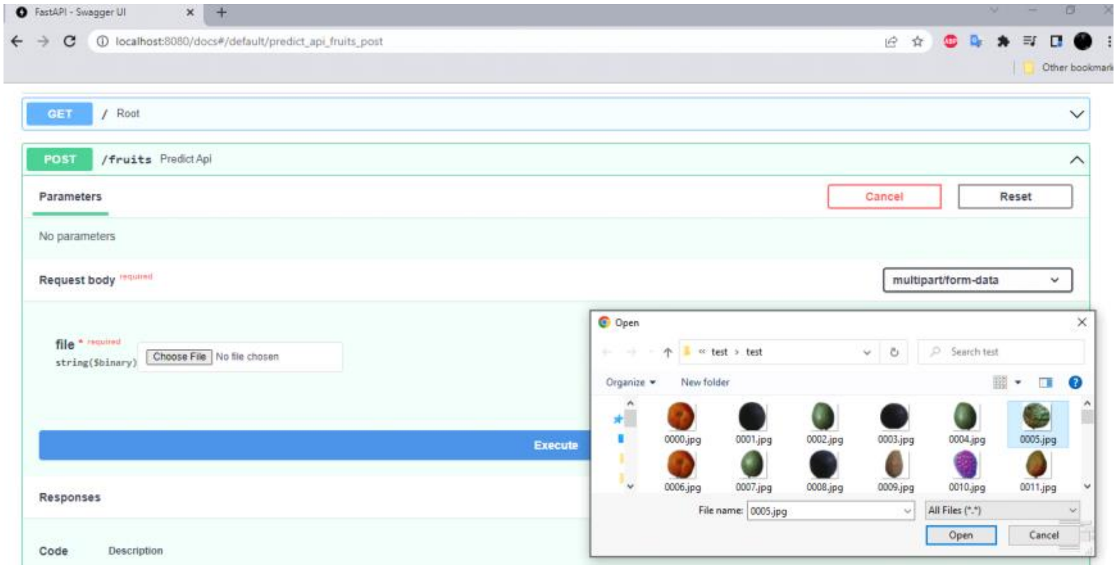


```
(.venv) vm-ubuntu@DESKTOP-1P02J1J:~$
(.venv) vm-ubuntu@DESKTOP-1P02J1J:~$ sudo docker build -f Dockerfile -t cnn_api_test .
[sudo] password for vm-ubuntu:
[+] Building 2.5s (13/13) FINISHED
 ⇒ [internal] load build definition from Dockerfile                                                                                             0.0s
 ⇒ ⇒ transferring dockerfile: 38B                                                                                                               0.0s
 ⇒ [internal] load .dockerignore                                                                                                                0.0s
 ⇒ ⇒ transferring context: 2B                                                                                                                   0.0s
 ⇒ [internal] load metadata for docker.io/library/python:3.8                                                                                    2.3s
 ⇒ [internal] load build context                                                                                                                0.0s
 ⇒ ⇒ transferring context: 192B                                                                                                                 0.0s
 ⇒ [1/8] FROM docker.io/library/python:3.8@sha256:2a01d88a1684e6d7f08030cf5ae73b536926c64076cab197e9e3d9f699255283                                0.0s
 ⇒ CACHED [2/8] WORKDIR /app                                                                                                                    0.0s
 ⇒ CACHED [3/8] COPY ./Pipfile /app/Pipfile                                                                                                     0.0s
 ⇒ CACHED [4/8] COPY ./Pipfile.lock /app/Pipfile.lock                                                                                           0.0s
 ⇒ CACHED [5/8] RUN pip install pipenv uvicorn cuda-python                                                                                      0.0s
 ⇒ CACHED [6/8] RUN pipenv sync                                                                                                                 0.0s
 ⇒ CACHED [7/8] COPY ./python_scripts/main.py /app/                                                                                             0.0s
 ⇒ CACHED [8/8] COPY ./python_scripts/trained_vgg16.ml /app/                                                                                    0.0s
 ⇒ exporting to image                                                                                                                           0.0s
 ⇒ ⇒ exporting layers                                                                                                                           0.0s
 ⇒ ⇒ writing image sha256:a85701f909a29a2659a8fc6995569d2620593654c73dd14258023045dbce4035                                                      0.0s
 ⇒ ⇒ naming to docker.io/library/cnn_api_test                                                                                                   0.0s
(.venv) vm-ubuntu@DESKTOP-1P02J1J:~$ docker run --rm -it --gpus all -p 8080:8080 cnn_api_test
Sequential(
  (0): Linear(in_features=25088, out_features=4096, bias=True)
  (1): ReLU(inplace=True)
  (2): Linear(in_features=4096, out_features=33, bias=True)
)
INFO:     Started server process [1]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
```

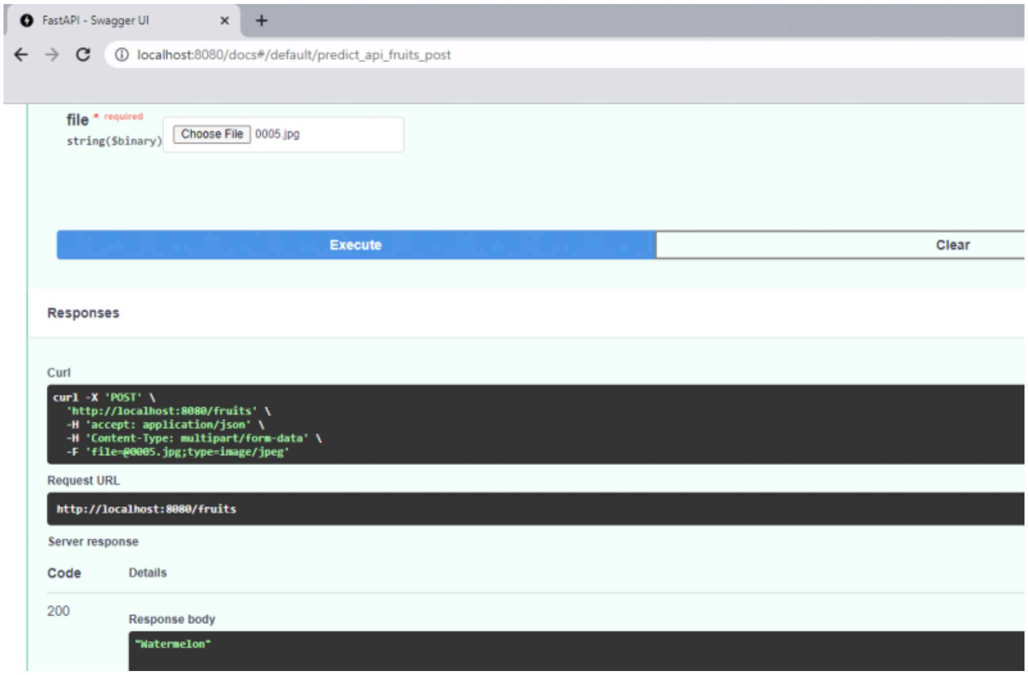
2. Open the FastAPI UI



3. Load a test image which will be passed to the model



4. Execute the script. Response body shows the class returned from the model



file * required
string($binary) Choose File 0005.jpg

Curl

```
curl -X 'POST' \
  'http://localhost:8080/fruits' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@0005.jpg;type=image/jpeg'
```

Request URL

```
http://localhost:8080/fruits
```

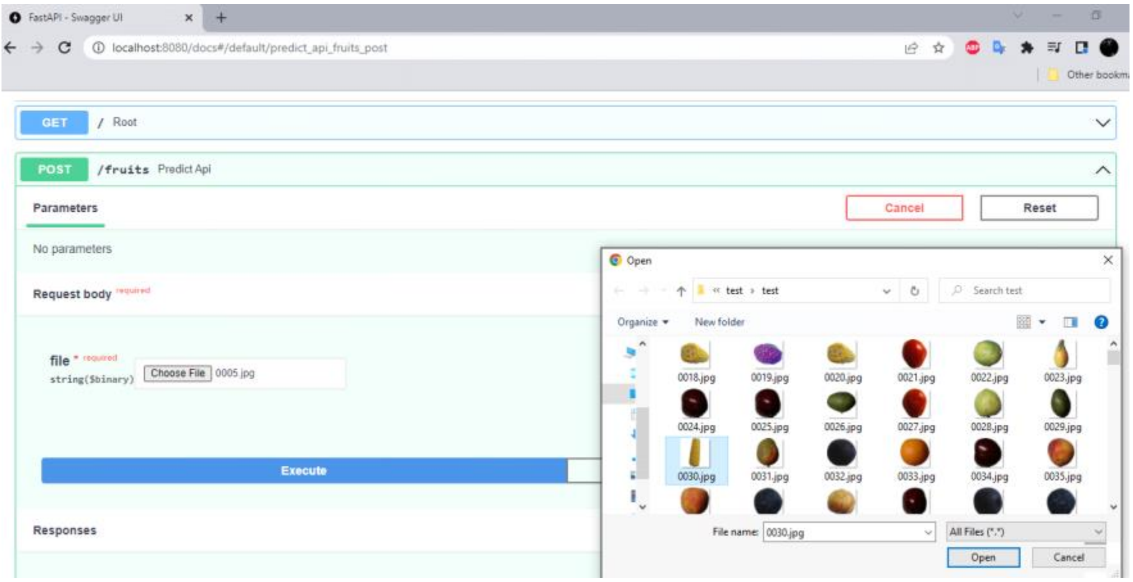
Server response

200

Response body

```
"Watermelon"
```

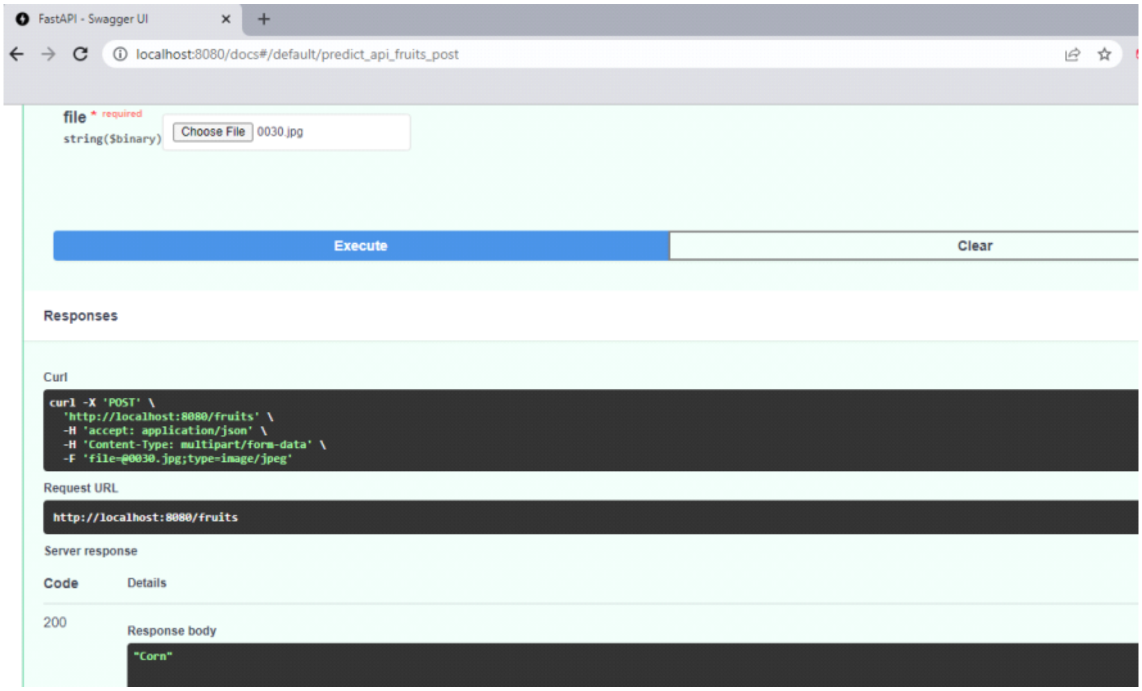
5. One more test image



No parameters

file * required
string($binary) Choose File 0005.jpg

File name: 0030.jpg
All Files (*.*)



0030.jpg

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/fruits' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@0030.jpg;type=image/jpeg'
```

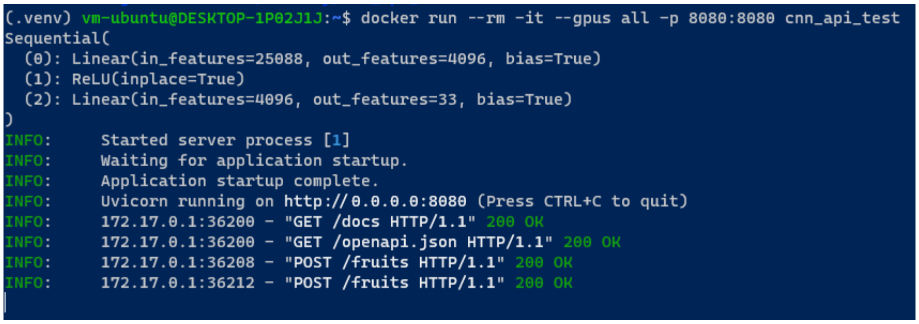
Request URL

```
http://localhost:8080/fruits
```

Server response

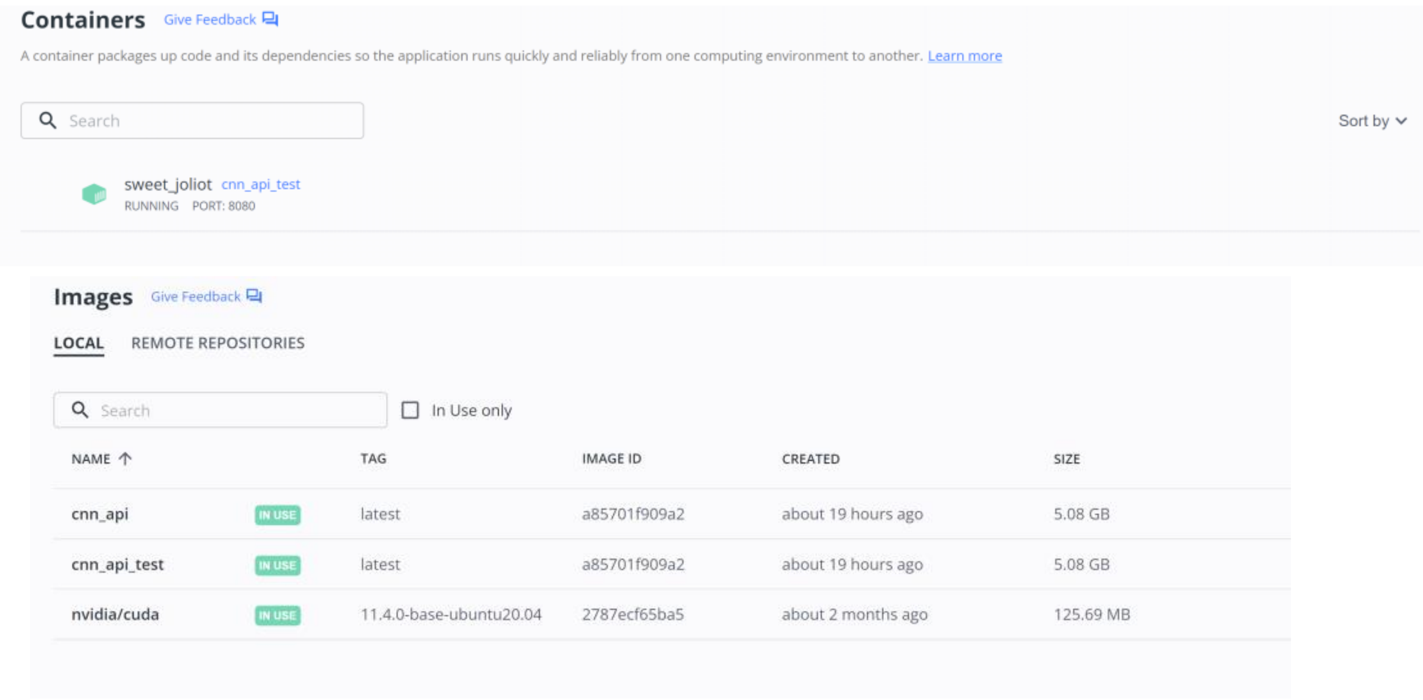
Code	Details
200	Response body "Corn"

Server responses



```
(.venv) vm-ubuntu@DESKTOP-1P02J1J:~$ docker run --rm -it --gpus all -p 8080:8080 cnn_api_test
Sequential(
  (0): Linear(in_features=25088, out_features=4096, bias=True)
  (1): ReLU(inplace=True)
  (2): Linear(in_features=4096, out_features=33, bias=True)
)
INFO:     Started server process [1]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
INFO:     172.17.0.1:36200 - "GET /docs HTTP/1.1" 200 OK
INFO:     172.17.0.1:36200 - "GET /openapi.json HTTP/1.1" 200 OK
INFO:     172.17.0.1:36208 - "POST /fruits HTTP/1.1" 200 OK
INFO:     172.17.0.1:36212 - "POST /fruits HTTP/1.1" 200 OK
```

Docker container and image



A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

sweet_joliot cnn_api_test
RUNNING PORT: 8080

□ In Use only

NAME ↑		TAG	IMAGE ID	CREATED	SIZE
cnn_api	IN USE	latest	a85701f909a2	about 19 hours ago	5.08 GB
cnn_api_test	IN USE	latest	a85701f909a2	about 19 hours ago	5.08 GB
nvidia/cuda	IN USE	11.4.0-base-ubuntu20.04	2787ecf65ba5	about 2 months ago	125.69 MB