**Code Lab Assignment 4**

*MSIS 2628*

Last updated: Oct 9, 2022

**Objective**

* Use docker-compose to build an ‘api’ service and a “consumer” service using Flask

**Technology**

* Docker
* Flask
* Python

**Process**

Overview

* We use [Flask](https://flask.palletsprojects.com/en/2.2.x/) to build a very small but runnable example - a meal recommendation service!
* Container 1, api, will publish meal recommendations via an API
* Container 2, consumer, will consume the meal recommendation and present it via HTML
* We organize our work in the following structure:

/

api/

code/

Dockerfile

consumer/

code/

Dockerfile

docker-compose.yml

Order

1. Build each container
2. Make sure that they’re able to talk to each other
3. Write the docker-compose file

Container 1 ‘api’

**Setup**

* Based on python
* Install the python library flask
* Create a volume called ‘app’
* Set this volume as your work directory
* Expose port 5000

**Run**

* Run the container interactively
* Share api/code directory

**Code**

* Create a file ‘api.py’ ([Starting point](https://pythonbasics.org/what-is-flask-python/))
* Make this file a minimalist API endpoint that randomly offers a random pick out of 15 meal recommendations along with a price
* The endpoint delivers 1 meal recommendation in JSON format

Container 2 ‘consumer’

**Setup**

* Based on python
* Install the python library flask
* Create a volume called ‘app’
* Set this volume as your work directory
* Expose port 80

**Run**

* Run the container interactively
* Share consumer/code directory

**Code**

* Create a file ‘consumer.py’ ([Starting point](https://pythonbasics.org/flask-http-methods/))
* Make this file a minimalist API consumer that displays the random meal recommendation along with the price (use a HTML template)

Bringing it together: Write an appropriate docker-compose.yml

* Make sure that ports and hostnames are part of the docker-compose.yml to avoid hard-coded dependencies.
* Give the containers appropriate names (e..g, api, consumer)
* Understand whether you have to create a [network](https://docs.docker.com/compose/networking/#specify-custom-networks) that allows the consumer to listen to the api

Submit

* Push results to github
* Submit the link to the repo on Camino

**Grading**

1. No hard-coded dependencies

<http://localhost:80> shows a formatted meal recommendation with an associated price.