Week 9 - Docker Lab

In this lab you will re-create the application as demonstrated in the module's videos.

Watch the videos and create a [Flask (Links to an external site.)](https://flask.palletsprojects.com/en/1.1.x/) container that call a [Redis (Links to an external site.)](https://redis.io/) container over a custom bridged [Docker (Links to an external site.)](https://docs.docker.com/network/bridge/) network. Initialize your Flask webserver to return an object from your individual project when the URL "**http://localhost:5000/data?key=<somevalue>**" is entered in Postman.

[Here](https://online.macomb.edu/courses/43853/files/13712706?wrap=1)[download](https://online.macomb.edu/courses/43853/files/13712706/download?download_frd=1)is a sample Python file to get you started.

----------------

Grading:

* A screen capture of the results of a Postman GET request from a Flask server running locally - 5pts
* A screen capture of the results of a Postman GET request from a Flask server in a Docker container - 5pts
* Your source code and docker files you developed for this assignment - 20pts

######

from flask import Flask, request, render\_template, jsonify

import redis

import json

######################

# For Local development...

# docker create --name local-redis -p 6379:6379  redis:latest

# For Docker...

# docker network create my-net

# docker create --name my-redis --network my-net -p 6379:6379  redis:latest

# docker create --name my-www   --network my-net -p 5000:5000  myweb:latest

######################

app = Flask(\_\_name\_\_)

######################

myRedis = None

######################

host = None

if \_\_name\_\_ == "\_\_main\_\_":

    host = "localhost"

elif \_\_name\_\_ == "app":

    host = "my-redis"

else:

    raise ValueError('Invalid "\_\_name\_\_ variable!')

######################

def InitRedisDatabase():

    key='key123'

    Item={

        "Value": "123",

        "Price": 2

    }

    global myRedis

    myRedis = redis.Redis(host=host, port=6379, db=0)

    myRedis.set(key, json.dumps(Item))

######################

@app.route("/data", methods=["GET"])

def GetData():

    if request.method == "GET":

        key = request.args.get("key")

        print("Key:", key)

        if key:

            data = myRedis.get(key)

            print("Data:", data)

            if data:

                return json.loads(data)

            else:

                return '{"Error":"Key could not be found!"}'

######################

InitRedisDatabase()

print(\_\_name\_\_)

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True, host=host, port=5000)