

## Attività 4.b: microservizi – REST API

### Esercitazione

**Obiettivo:** avviare un microservizio in un container che fornisca una REST API per l'acquisizione dei dati della tabella *corsi*

#### Fase 1: prova dell'applicazione

1. Per poter provare l'applicazione prima di creare l'immagine è necessario aver installato nella macchina host *Python*, *flask* e *mysql-connector-python*. (attenzione: nessun container deve essere attivo)
2. Creare una cartella *muscolin* ed una sotto cartella *api*
3. Nella cartella *api* creare l'applicazione flask (*apiApp.py*) di esempio (dopo aver scoperto l'indirizzo del container *mdb*, ad esempio con l'inspect della rete bridge) (per la copia dei file sulla macchina virtuale vedi la nota in fondo)

```
from flask import Flask, jsonify
import mysql.connector

user="root"
password="Password123!"
host="172.26.0.2"          ←===== indirizzo mdb
port=3306
database="muscolin"

app = Flask(__name__)

@app.route('/corsi')
def elencoCorsi():
    connessione = mysql.connector.connect(
        user=user,
        password=password,
        host=host,
        port=port,
        database=database
    )

    cursore=connessione.cursor(dictionary=True)
    cursore.execute("SELECT * FROM corsi")
    risultato=cursore.fetchall()
    cursore.close()
    connessione.close()

    risposta=jsonify(risultato)
    risposta.headers.add("Access-Control-Allow-Origin", "")
    return(risposta)
```

4. Provare l'applicazione sulla macchina host, accedendo da un browser esterno o con curl

#### Fase 2: preparazione dell'immagine

1. Generare l'immagine *api* (basata sull'immagine *python*), specificando, se necessario, il proxy (vedi attività 3.d) ed installando nell'immagine *flask* e *mysql-connector-python*

```
from python
run python3 -m pip install mysql-connector-python==8.0.29
run python3 -m pip install flask
copy . /
```

```
cmd python3 -m flask --app apiApp.py run --host 0.0.0.0
```

2. Avviare il container (con nome *api*) e mappare la porta 5000

```
$ docker container run --name api --rm -d -p 5000:5000 api
```

3. Provare da un browser; in caso di problemi lanciare i container in modalità interattiva e verificare i messaggi

**NOTA: copia dei file fra la macchina fisica e la macchina virtuale (comandi da lanciare da Windows)**

- Copia da Windows sulla homedir della macchina virtuale (attenzione al punto)

```
$ scp fileSuWindows tps@xxx.xxx.xxx.xxx:.
```

- Copia dalla macchina virtuale alla cartella corrente su Windows (attenzione al punto)

```
$ scp tps@xxx.xxx.xxx.xxx:cartellaLinux/fileLinux .
```