

Codio: Unit 3

Practical Activity: Python Virtualisation

For the practical assessment (assessment 2) you are required to create an application based on Micro – Services. As discussed in LC1, Micro-Services use a different paradigm to create applications than the older monolithic approach. Where a monolithic application could pass data via shared (or global) variables (as all modules were effectively in the same address space) Micro-Service Applications (MSA) cannot. The current approach to developing Micro-Service apps is that each application represents an individual business problem or process. Most developers create each MSA in its own container (e.g. Docker or LXC) and communication between apps is via messages and API (RESTful APIs are very common). Unfortunately, if you try and use containers on Codio, you will fail as each Codio session is already a container – and you cannot run containers within containers.

There are a variety of options you can utilise to get around this and these are discussed below:

1. Use virtualenv or venv
2. Use Conda
3. Use VSCode DevContainers

Exercise

Use the Venv instruction to set up a Python web server in Codio.

A. Log into Codio and update the environment: type

```
sudo apt-get update
```

B. Install the venv package with

```
sudo apt-get install Python3-venv
```

C. Type the following command to create a venv:

```
python3 -m venv websites/test/
```

D. cd to the test directory and run the webserver command

```
python3 -m http.server
```

E. Create a new file called getresp.py and enter the code below:

```
import http.client

con = http.client.HTTPConnection("localhost", 8000, timeout=10)

con.request("GET", "/")

r1 = con.getresponse()

print(r1.status, r1.reason)
```

F. Run this file using

```
python3 getresp.py
```

You should get the standard response below:

```
200 OK
```

Output

```
Lauxton@pg7700x6750xt:~$ python3 -m venv websites/test/
Lauxton@pg7700x6750xt:~$ cd websites/test
Lauxton@pg7700x6750xt:~/websites/test$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [05/Oct/2023 19:01:04] "GET / HTTP/1.1" 200 -

```

```
Lauxton@pg7700x6750xt:~/websites$ vln getresp.py
Lauxton@pg7700x6750xt:~/websites$ python3 getresp.py
200 OK
Lauxton@pg7700x6750xt:~/websites$
```

```
1 import http.client
2
3 con = http.client.HTTPConnection("localhost", 8000, timeout=10)
4
5 con.request("GET", "/")
6
7 r1 = con.getresponse()
8
9 print(r1.status, r1.reason)
```