

## Developing an API for a Distributed Environment

In this session, you will create a RESTful API which can be used to create and delete user records. Responses to the questions should be recorded in your e-portfolio.

You are advised to use these techniques to create an API for your team's submission in Unit 6. Remember that you can arrange a session with the tutor during office hours for more support, if required.

Using the [Jupyter Notebook workspace](#), create a file named api.py and copy the following code into it (a copy is provided for upload to Codio/GitHub): You can [install Jupyter Notebook on your local machine following these instructions](#) or via the [University of Essex Software Hub](#).

#source of code: <https://codeburst.io/this-is-how-easy-it-is-to-create-a-rest-api-8a25122ab1f3>

```
from flask import Flask
from flask_restful import Api, Resource, reqparse

app = Flask(__name__)
api = Api(app)

users = [
    {
        "name": "James",
        "age": 30,
        "occupation": "Network Engineer"
    },
    {
        "name": "Ann",
        "age": 32,
        "occupation": "Doctor"
    },
    {
        "name": "Jason",
        "age": 22,
        "occupation": "Web Developer"
    }
]

class User(Resource):
    def get(self, name):
        for user in users:
            if(name == user["name"]):
                return user, 200
        return "User not found", 404

    def post(self, name):
        parser = reqparse.RequestParser()
        parser.add_argument("age")
        parser.add_argument("occupation")
        args = parser.parse_args()

        for user in users:
            if(name == user["name"]):
```

```
        return "User with name {} already exists".format(name), 400

    user = {
        "name": name,
        "age": args["age"],
        "occupation": args["occupation"]
    }
    users.append(user)
    return user, 201

def put(self, name):
    parser = reqparse.RequestParser()
    parser.add_argument("age")
    parser.add_argument("occupation")
    args = parser.parse_args()

    for user in users:
        if(name == user["name"]):
            user["age"] = args["age"]
            user["occupation"] = args["occupation"]
            return user, 200

    user = {
        "name": name,
        "age": args["age"],
        "occupation": args["occupation"]
    }
    users.append(user)
    return user, 201

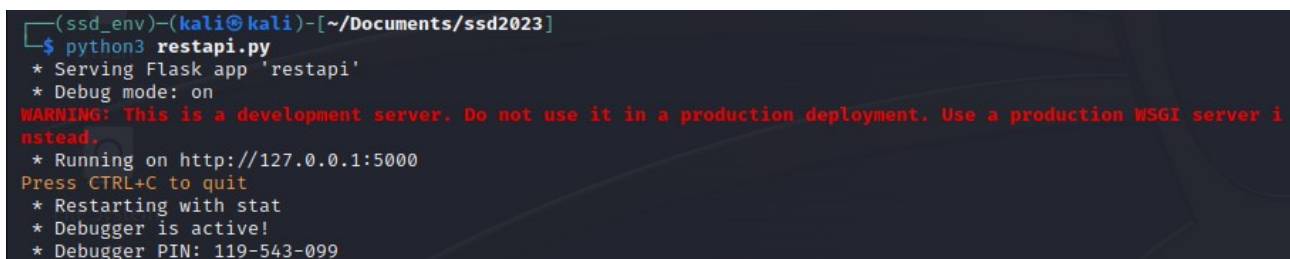
def delete(self, name):
    global users
    users = [user for user in users if user["name"] != name]
    return "{} is deleted.".format(name), 200

api.add_resource(User, "/user/")

app.run(debug=True)
```

## Question 1

Run the API.py code. Take a screenshot of the terminal output. What command did you use to run the code?



```
(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ python3 restapi.py
* Serving Flask app 'restapi'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 119-543-099
```

I used `python3 restapi.py` to run the code on my own env as flask was unsupported on the codio workbook.

## Question 2

Run the following command at the terminal prompt: w3m <http://127.0.0.1:5000/user/Ann>

What happens when this command is run, and why?

```
(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ python3 restapi.py
* Serving Flask app 'restapi'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 119-543-099
127.0.0.1 - - [14/Apr/2023 01:29:15] "GET /user/Ann HTTP/1.1" 404 -
```

I was unable to get a 200 – OK response from the code provided. I used Insomnia as a API tester and it reported that the server was not responding.

## Question 3

Run the following command at the terminal prompt: w3m <http://127.0.0.1:5000/user/Adam>

What happens when this command is run, and why?

```
(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ python3 restapi.py
* Serving Flask app 'restapi'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 119-543-099
127.0.0.1 - - [14/Apr/2023 01:29:15] "GET /user/Ann HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:30:11] "GET /user/Adam HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:32:04] "GET /user/ann HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:32:16] "GET /user/Ann HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:32:24] "GET /user/Adam HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:33:37] "GET /users/Adam HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:33:46] "GET /Adam HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:33:55] "GET /user/Adam HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:34:15] "POST /user/Adam HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:36:19] "GET /user/Jason HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:36:23] "GET /user/James HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:36:28] "GET /user/Ann HTTP/1.1" 404 -
^C
```

Here the code is technically correct – there is no ‘Adam’ in the user list so the output should return a 404 – NOT FOUND. But, as is documented, all users were 404, and the same message concerning the server popped up again on Insomnia.

```

(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ pylint restapi.py
***** Module restapi
restapi.py:11:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:29:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:33:0: C0325: Unnecessary parens after 'if' keyword (superfluous-parens)
restapi.py:36:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:42:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:44:0: C0325: Unnecessary parens after 'if' keyword (superfluous-parens)
restapi.py:46:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:54:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:60:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:62:0: C0325: Unnecessary parens after 'if' keyword (superfluous-parens)
restapi.py:66:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:74:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:79:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:81:0: C0303: Trailing whitespace (trailing-whitespace)
restapi.py:1:0: C0114: Missing module docstring (missing-module-docstring)
restapi.py:4:0: W0105: String statement has no effect (pointless-string-statement)
restapi.py:30:0: C0115: Missing class docstring (missing-class-docstring)
restapi.py:31:4: C0116: Missing function or method docstring (missing-function-docstring)
restapi.py:37:4: C0116: Missing function or method docstring (missing-function-docstring)
restapi.py:45:23: C0209: Formatting a regular string which could be a f-string (consider-using-f-string)
restapi.py:55:4: C0116: Missing function or method docstring (missing-function-docstring)
restapi.py:75:4: C0116: Missing function or method docstring (missing-function-docstring)
restapi.py:76:8: W0603: Using the global statement (global-statement)
restapi.py:78:15: C0209: Formatting a regular string which could be a f-string (consider-using-f-string)

```

Using pylint I tried to clean up the code to see if that would improve the output, and did so up to the level that functionality would allow:

```

(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ pylint restapi.py
***** Module restapi
restapi.py:43:23: C0209: Formatting a regular string which could be a f-string (consider-using-f-string)
restapi.py:75:8: W0603: Using the global statement (global-statement)
restapi.py:77:15: C0209: Formatting a regular string which could be a f-string (consider-using-f-string)

Your code has been rated at 9.27/10 (previous run: 8.54/10, +0.73)

(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ python3 restapi.py
* Serving Flask app 'restapi'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 119-543-099
127.0.0.1 - - [14/Apr/2023 01:55:46] "GET /user/Ann HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:56:02] "GET /user/Jam HTTP/1.1" 404 -
127.0.0.1 - - [14/Apr/2023 01:56:11] "GET /user/Adam HTTP/1.1" 404 -

```

But I still received the same errors as before, even though the code matched the website's description.

#### Question 4

What capability is achieved by the flask library?

Clearly Flask allows the creation, updating, and deletion of users and user categories on an application.

It may have run into errors because it did not have a complementary json file.

Though the logic of the code is clear, I could not make it work – which may mean that my coding chops are not up to snuff.

And unfortunately, since this is a compressed module, I did not have the time I would have liked to dedicate to figure it out. So for the final project, another API might need to be used.