Python API ZTP Exercise

Objective

The main goal of this exercise is to write Restful API Service and client which allow machines to register to service and the service manager to retrieve the IP address of the registered machines using standard http methods for RESTFUL.

Server

The server has 3 API's Registration for receiving client calls , Retrieval for communication with the service management and Management for administration those API's are defined below. The server after client's registration triggers an API call to the service manager.

Client

Client issues Register API call every time it runs. in this call the client sends to the server devices id (MAC Address or CPU ID) and it's IP address.

Server API Definitions

Registration

Register
url: /ztp/register/{id}
Http method: put
Payload: Json

```
{ "ip": "1.1.1.1",
  "name":"",
  "userdata":"" }
```

Retrieval

Check
url: /ztp/check/{id}
Http method: get
Payload: Json

```
{ "ip": "1.1.1.1",
  "name":"",
  "userdata":"" }
```

Management

List

url: /ztp/mng/list/
Http method: get
Payload: Json
{ hosts: [id,id,id] }

Delete API
url /ztp/mng/{id}/delete
Http method: get

Edit API
url /ztp/mng/{id}
Http method: put
Payload: Json
```
{ "ip": "1.1.1.1",
  "name":"",
  "userdata":"" }
```

Trigger API call

Issues after Registration call
http://adress/api/exec?ip={ip}

Requirements

This exercise should be implemented using the Python programming language (2.x or 3.x) and Flask framework and need to take care of the following:

1. Implemented using Python best practices
2. Implemented using PEP8 standard
3. Implemented the API as illustrated in the API definitions section
4. Implement and add unit tests for the code
5. Run code against PEP8 to make sure it is written as expected
6. The code should be hosted in private github repository
7. **No need to have a user management functionality but the code should be written in a way that would not prevent adding it in the future**
8. **No need to have encryption and client verification but the code should be written in a way that would not prevent adding it in the future**