

# UberServer homework

## Introduction

Imagine that your new boss is on his way out for a weekend fishing trip. Before he leaves, he asks you to quickly implement a critical piece needed to deliver for a huge new deal. He needs a server that can both return current time and return a point-in-time location of a VIP (we can't tell you the name) whose movements we've been tracking.

## Endpoints

The server must support following use cases:

### GET /v1/now

Endpoint should return current UNIX epoch timestamp in ISO 8601 format in json object in field `now`.

Example response:

```
{
  "now": "2020-12-07T09:05:29+0000"
}
```

### GET /v1/VIP/{point-in-time}

Endpoint should return coordinates of the mysterious person at `point-in-time` where `point-in-time` is a Integer.

Example response:

```
{
  "source": "vip-db",
  "gpsCoords": {
    "lat": value,
    "long": value
  }
}
```

## SLA

Clients that will call your service are impatient however. And they will sue us if it takes more than 5 seconds for the service to respond. They are ok with us failing. But fail fast.

## Backend

The location of person is stored in a database that exposes the data via HTTP interface. Luckily for you, it can be summoned via `docker-compose.yml` found in this directory. If you run the `docker-compose up`, the service should start listening on `localhost:8088`. Your new http server should then obtain the coordinates of the VIP by calling `GET localhost:8088/v1/coords/{point-in-time}`.

Example request:

```
$ curl -X GET localhost:8088/v1/coords/2
{"type":"coords","latitude":24.77494,"longitude":-165.51148}
```

But be warned. The database is not great and sometimes fails, responds with unexpected status code or takes a long time to respond. Keep that in mind.

## Stack

You can use any Python framework of your choice and if there are too many good options to select from you may try to play with FastAPI that is very popular nowadays.

You are also very welcome to use any other 3rd party libraries. We want to see the best you can do.

## Quality

This service is of big importance so think about how will we be able to verify that it's working correctly in the production. Think about what are important things to log, what metrics might be interesting, how you handle failures etc. It's up to you how deep you want to go but this is your chance to WOW us :)

## Bonus

Extend server with authentication of your choice and make at least `GET /v1/VIP/{point-in-time}` protected against use by not authenticated users.

List of allowed users can be hardcoded in code or stored in database of your choice(if you are planing to use database please extend the configuration of `docker-compose.yml` in such way that we are able to test the solution in same way as if database was not involved).

***PS:** We don't expect this task to take more than 4 hours of your time. Although you can take as much time as you need.*