

Python Software Engineer Assignment

Before being invited for an interview with Pole Star, you will need to do according assignment and send it back for review by our Development Team.

It should not take you longer than a few hours. Please get comfortable, read the requirements carefully, take your time to think about the best solution you can come with, and then proceed with the implementation.

Introduction

Here at Pole Star, we track all kind of vessels and ships from every flag around the World. Every ship is uniquely identified with a 7-digit number assigned by the International Maritime Organisation, the *IMO number*.

We receive and process geographical positioning signals from satellite devices on thousands of vessels per minute. Each one of those positions is stored, together with its timestamp, and related to a ship in our databases. This gives us a good tracking history of every vessel enabling the business to perform further analysis on them.

Assignment

Given data

We will give you a CSV file which contains 2000 geographical positions related to 3 different ships that are currently sailing the oceans. The CSV file will look like this:

```
9595321,2019-01-14 19:05:32+00,18.4211502075195,-64.6109008789062
9632179,2019-01-14 18:59:06+00,49.9175834655762,-2.40604996681213
9595321,2019-01-14 18:47:31+00,18.4211502075195,-64.6109008789062
9632179,2019-01-14 18:40:18+00,49.8978996276855,-2.51836657524109
9247455,2019-01-14 18:31:22+00,17.5403633117676, 69.7120666503906
...
```

As you can see, it is formed by 4 columns separated by commas. The first column is the **IMO number**, second is the **timestamp** where the position was taken, and third and fourth are the **latitude** and **longitude**, respectively.

The ships for which the positioning data is for are these three ships:

- **Mathilde Maersk**, IMO number **9632179**.
- **Australian Spirit**, IMO number **9247455**.
- **MSC Preziosa**, IMO number **9595321**.

Tasks to do

Given the data in the CSV file provided to you:

1. Design a **relational database** schema to store the data for the Ships and their Positions.
2. Write Python code to automatically load the CSV data into a relational database.
3. Write, using Python, a REST API that implements 2 endpoints:
 - a. `/api/ships/`: must show a list of current ships in the database. Payload should show *at least* the `imo` and `name` of every ship record.
 - b. `/api/positions/<imo>/`: given the `imo` of one of the ships, must show the positions related to that ship in descendent order (starting from the last one received, list all of them until the oldest one). Payload should also show *at least* the `latitude` and `longitude` of every position.

Extra tasks (not mandatory)

4. If you have extra time and want to see a graphical representation of your API being used, you can trigger a development server for your code and open in a browser the given `index.html` file. Feel free to modify it if you need to. It should look as follows:

Python Assignment



5. If you know how to use Docker and working with containerized environments is something usual for you, please go ahead and do it (**very** appreciated) including, if you want, a UI for the final application using the provided `index.html` file (**brilliant**).

Requirements

- Language used must be **Python**, latest version (**3.7** or **3.6**).
- You are **absolutely free to choose any framework** or library you prefer to implement your API. In Pole Star we daily use Django/DRF and Flask, depending on the project, but you can pick up anything you like for this assignment.
- You are also **free to choose any relational database** you want. We always use PostgreSQL, but SQLite might be used to ease this assignment if you prefer, for example.
- Your code **must contain tests** to prove it works as expected. Testing framework is of free choice as well (Django `django.test.TestCase`, Pytest, Nose... the one that makes your life easier).
- Your code **must run standalone** out of the box in our local environments. It must **not need** to install anything else, but a Python virtual environment *as much*. Again, if you provide us with a Docker environment it will be **greatly** appreciated.
- Whatever you prefer to do, you **must include a README** to tell us how to run your API.
- Think before write code. There is no timing in this exercise, it can take as long as you need, once started.

Good luck, and hope to see you soon!