SKURT

Programming Challenge

Context:
Skurt Engineering is working on a new technology called Skurt Kit that will give us the ability to geolocate cars in our fleet at any given time. Being able to obtain the location and status of a particular car is also extremely valuable since this would help us with the safety of our cars and also increase the efficiency of our deliveries. For example, we can deliver cars in a very cost-efficient manner by finding the car closest to the customer based on the location of their device.

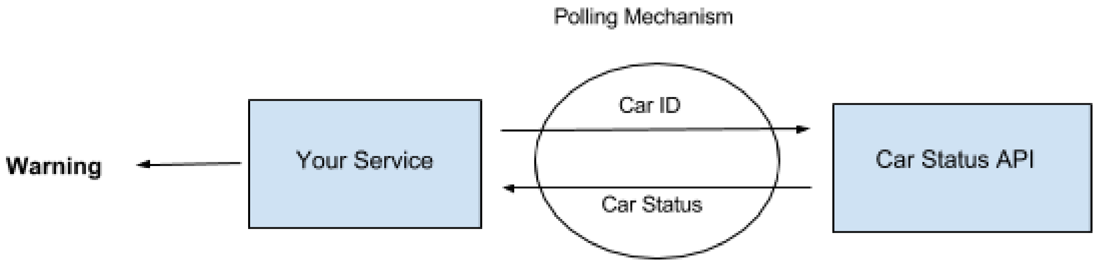
Task:
The first use case for this technology is a security feature that will alert us within 5 minutes of a car being driven out of a zone predefined by us. Your task is to implement this.

You should poll the Car Status API for status updates on a given set of cars. Anytime a car goes out of range your service should send an email to engineering@skurtapp.com with the missing car's ID. We will provide you with a list of 10 unique car IDs that you can assume are valid and exist in our system.

NOTE: There are actually 11 cars in the system: Car ID # 11 is a fake we put in the system for debugging purposes. It is always out of the bounding zone and can be used to test. It's position **never** changes.

Write a service that monitors the status of each of these cars and alerts us via email if any car is out of their designated zone. The monitoring interval is up to you- find a good balance between not overloading the API and the accuracy of the warning (in terms of how fast we are alerted).

Proposed Architecture:



Car Status API Documentation:

URL: http://skurt-interview-api.herokuapp.com/

- GET /carStatus/{CarID}
  - @param carID: An integer between 1-11 that represents a car's ID. Cars with ID 1-10 are valid cars whose position keep changing. Car #11 is a test car that is always out of the bounding zone.
  - Returns GeoJSON with a car's status (current position as a coordinate)

Requirements:

- **Polling mechanism** to the car status API
- Warning signaled within 5 minutes
- No restrictions on programming language
- A car on the boundary line is **still** in the zone
- The code is well tested
- If the endpoint does not return for some reason- make sure to **alert** us!

Helpful Resources

- geojson.io: This will be very useful with debugging. Copy paste the GeoJSON (equivalent to JSON) returned by the endpoints entirely into the right panel of the site to view the location of the car.
- http://geojson.org/: If you're interested in the specification and details of geojson.

Submission:

Please email your solution to engineering@skurt.com

Let us know if you have any questions!