Frotcom Challenge

There is a non profit organization that performs **analytics** operations over tracking data to find tendencies and singularities. Frotcom is willing to participate in the project by providing **anonymous** data collected from the tracked vehicles.

In this experiment, only the data for the portuguese territory will be used. For performance sake, the tracking data of **each vehicle** should be accumulated and sent in batches of at least **100 packets**.

Each vehicle has a Frotcom tracking device that collects location and speed data and sends it to a queue in **Microsoft Azure**. Assume that one queue holds the tracking data of all vehicles. The data model can be found in *Frotcom.Challenge.Data.Models.Packet*.

The goal of this challenge is to create a .NET Core console application that will consume the tracking data queue and apply the rules described above. Since this is just an exercise, the data sending to the API will be replaced by a log, for example:

```
12/17/2020 4:28:04 PM: Vehicle 10 sent 100 packets in Portugal
12/17/2020 4:28:04 PM: Vehicle 37 sent 100 packets in Portugal
12/17/2020 4:28:04 PM: Vehicle 34 sent 100 packets in Portugal
12/17/2020 4:28:04 PM: Vehicle 30 sent 100 packets in Portugal
12/17/2020 4:28:04 PM: Total: 15000, InPortugal: 2460
12/17/2020 4:28:04 PM: Vehicle 22 sent 100 packets in Portugal
12/17/2020 4:28:04 PM: Vehicle 41 sent 100 packets in Portugal
12/17/2020 4:28:04 PM: Vehicle 3 sent 100 packets in Portugal
```

Requirements:

- Console application where it should be possible to *Start* and *Stop* the process.
- Read the vehicles tracking data from a queue, using *Frotcom.Challenge.Queue*.
- Use *Frotcom.Challenge.Reverse.Geocoding* to get the country for a given latitude and longitude.
- Send batches of 100 tracking packets per vehicle to the client API (endpoint “v2/vehicle/{vehicleId}/data”, POST method). For this exercise log to console instead, but the effort to change to send to the API should be minimum.
- Log, every 10 seconds, the total packets received and the total packets received inside Portugal.
- Take advantage of asynchronous and parallel programming, having in consideration possible concurrency problems.
- Code readability and performance are important factors.

Implement your solution inside *Frotcom.Challenge.SendTrackingDataWorker*, feel free to apply any change on the existing libraries.