|  |  |
| --- | --- |
| Vehicles Status Report  Stockholm |  |

Requirement Definitions for the application functionality

Analysis of the Problem

|  |  |
| --- | --- |
| Date | Services Expected By Application: |
| May 3, 2018 | Vehicles Status Report  Stockholm |

Respective Company has a number of connected vehicles that belongs to a number of customers. They have a need to be able to view the status of the connection among these vehicles on a monitoring display. The vehicles send the status of the connection one time per minute. The status can be compared with a ping (network trace); no request from the vehicle means no connection.

# Features expected/ identified

Web GUI (Single Page Application Framework/Platform). - An overview of all vehicles should be visible on one page (full-screen display), together with their status. –

* It should be able to filter, to only show vehicles for a specific customer. - It should be able to filter, to only show vehicles that have a specific status.
* 2. Random simulation to vehicles status sending.
* 3. If database design will consume a lot of time, use data in-memory representation.
* 4. Unit Testing.
* 5. .NET Core, Java or any native language.
* 6. Complete analysis for the problem.

# Architectural design view

From the problem definition, company wants to track the status of the vehicles every minute, We need a UI to display, we can choose from our latest stack, We can choose Angular as a front end SPA technology.

We need a centralized system to which vehicles can update their current status and front end can also request the status every minute from the same, we can choose a technology which is stable and sufficient for this piece of task in terms of better reuest/response communications with not very monolithic requirements.

We can choose Dropwizard technology to implement a lightweight microservice which has support of jersy as rest , jettey as embedded server which can be UP and running very quickly.

Also this microservice can be dockerized and can be made available to several virtual machine & containers to make the system highly available for the vehicles interacting at any time either by portable device mobile or from their screen embedded into their cars may be.

# Analysis behind the solution design

Analysis has been done keeping in mind the report should be helpful for the company to track the current status to avoid frauds & misuses of the company agreements, legal or asset theft cases.

**UI SELECTION:**

A Realtime refresh of the status list is required to have accurate information in the screen all time whenever management want to refer the screen. Angular technology provide good look and feel and responsive APIs that can be used to develop the interface quickly and can be easily be deployed to clients location with minimal CLI commands provided. Angular also integrated with Reactive library that helps subscribe the data as and when needed. Angular technology as UI should be best candidate fit for the requirements. Full screen library along with UI components are rich & responsive , also it provides a clean separation between the gui components interacting with its component coed platform and if needed to call to other backend API then we it provides service to have a clean isolated layer architecture. Filter functionality can be developed using the events from screen to components and filter the objects having the status value or the particular customer name. Anglar 2 way quick binding helps in displaying any flag or error quickly.

**µService SELECTION:**

Dropwizard should be best fit for providing lightweight embedded quick up and running , have matrix and health check support , best for rest implementations. Can be dockerized to be deployed to several virtual machines. Dropwizard provides supports for ORM support/hibernate, db migration ,testing..etc

**Lombok:** API can be used to avoid java noises(geter/setter) in the implementation, its builder feature can be used for populating the objects from different layers.

**DB/Data stores:**

The real objects can be identified as CUSTOMER, VEHCILE, ADDRESS and corresponding database tables can be created.

Mongo/Postgre/Mysql can be used for the same.

In memory **H2** database is perfect candidate to complete local developments.

**UNIT Tests/TDD :** JUNIT with Mockito /power Mockito for static codes can be used.

**Build Tool**: Maven/Gradle is the candidates.

**IDE Selection**: **InteliJ**, love to code & code is life with IntelIJ/WebStorm for Angular/SPA development.

# Deliverable Materials

1. Micro Service application full code.
2. SPA angular code

Git hub:

<https://github.com/jaiprakashpandey/polten>

# Build Deployment Steps

Here are the deployment steps:

**µService application:**

1. **Download/clone/Unzip the project received if needed to maintain and develop else direct executable jar file along with a resource file names “service.yaml” can be started on server from steps 3.**
2. **Go to root directory, maven clean install , will create the µService executable jar**
3. **Start the DB Setup with below command**

**java -jar vehicle-status-1.0-SNAPSHOT.jar db migrate C:\jai\ang\alten\vehicleStatus\src\main\resources\service.yaml**

**(IN memory DB has been used for this implementation)**

1. **Start the server :**

**java -jar vehicle-status-1.0-SNAPSHOT.jar server C:\jai\ang\alten\vehicleStatus\src\main\resources\service.yaml**

**Note: please use the paths according to local servers**

**SPA Angular front end:**

1. **Install Angular-cli-ghpages if interested to run from code from git hub**

**Npm install -g angular-cli-ghpages**

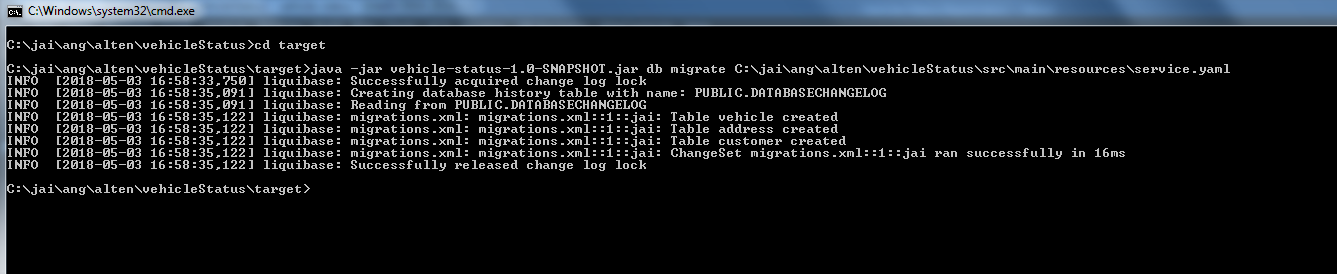
1. **Ng build –prod -base-href=”** **https://github.com/jaiprakashpandey/polten.git”**
2. Start server :

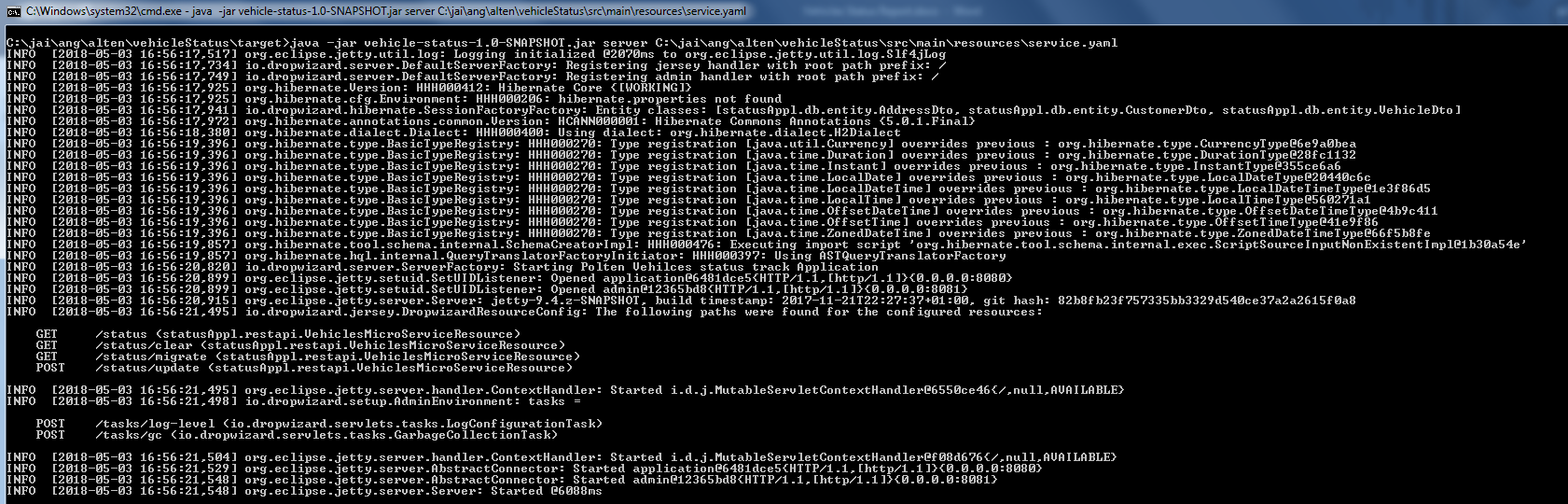
angular-cli-ghpages

1. Access the application using localhost:4200/

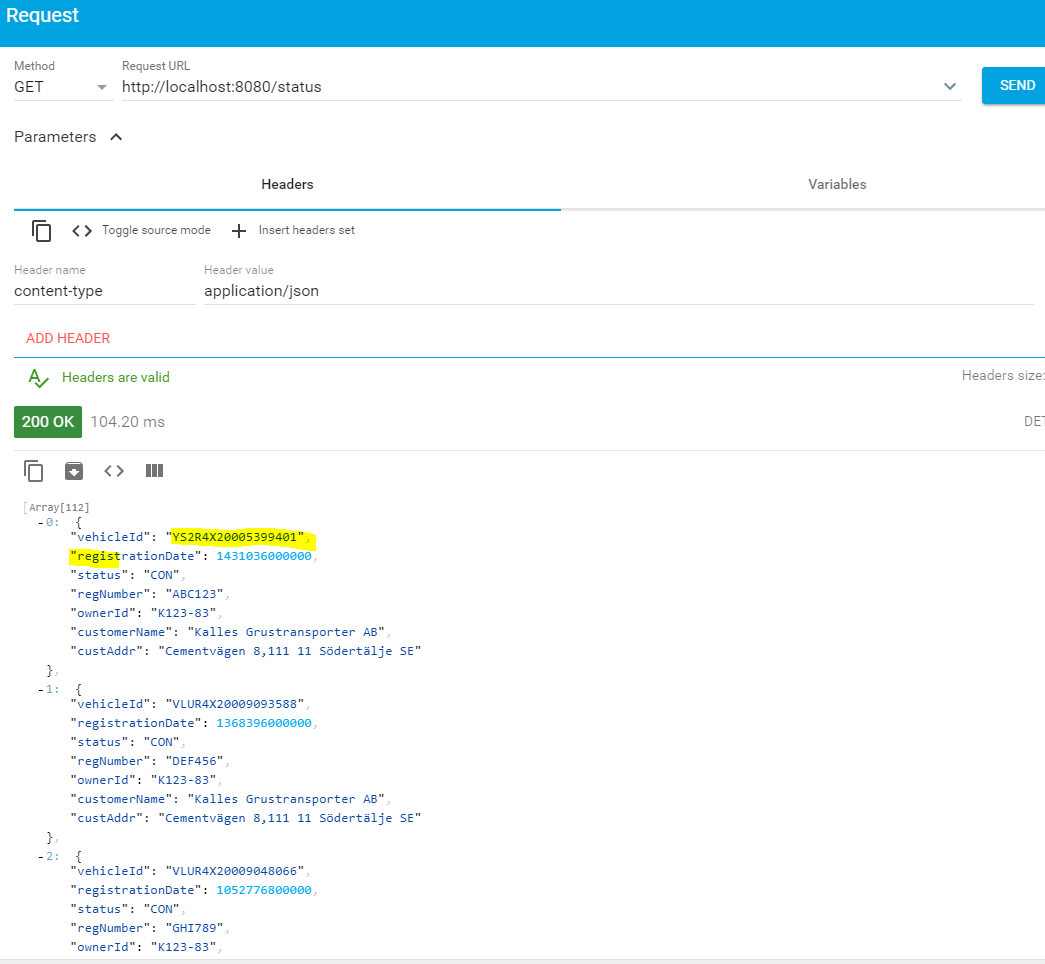
# Some snaps of the application running

DB Migration/Setup: & Server starting:





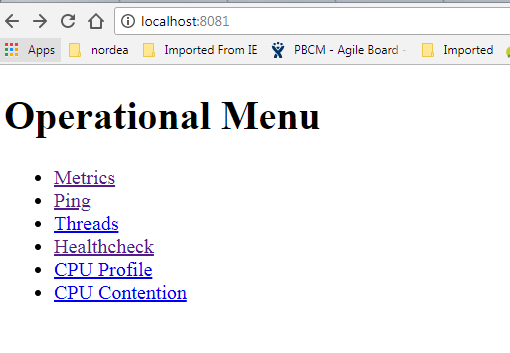
Backend microservice testing



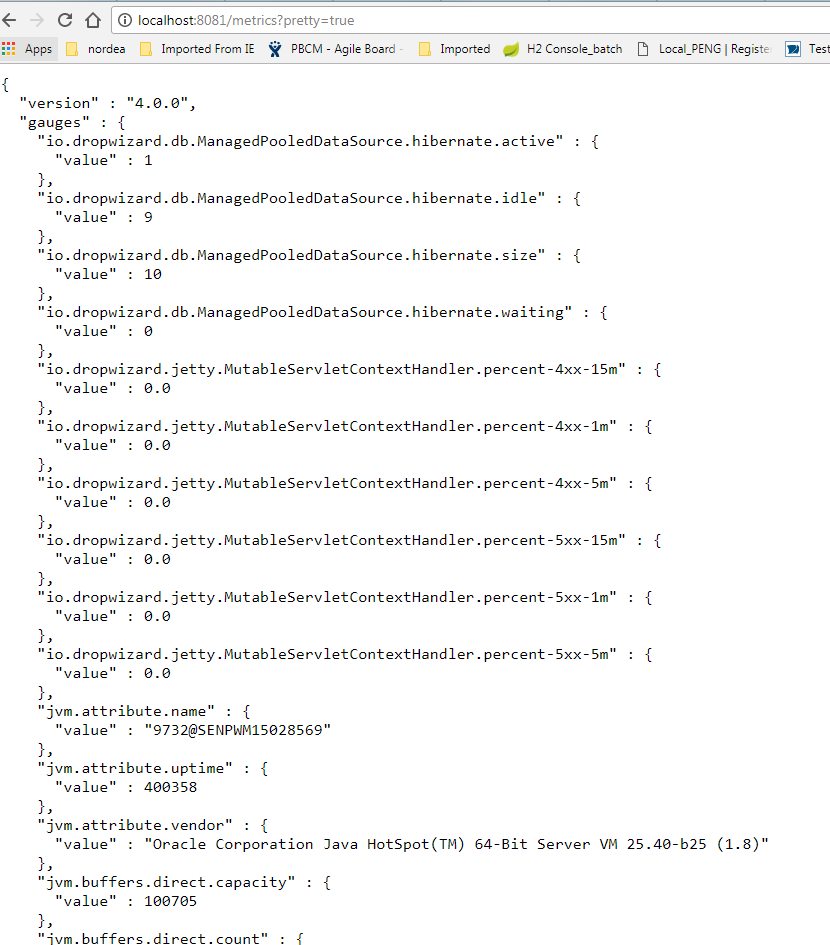
# Health Checks /Metrics results

Health check Admin URL:

http://localhost:8081/

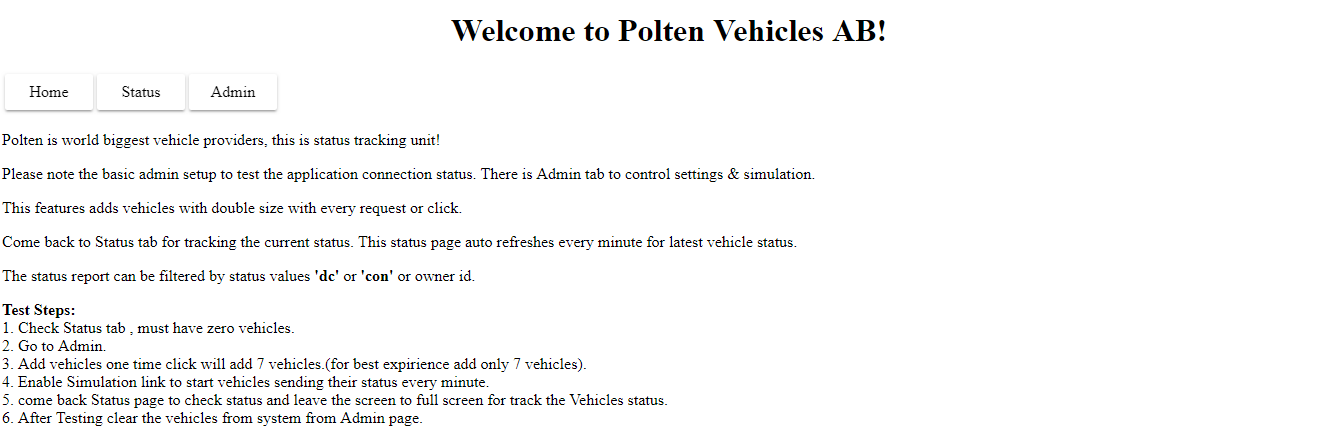




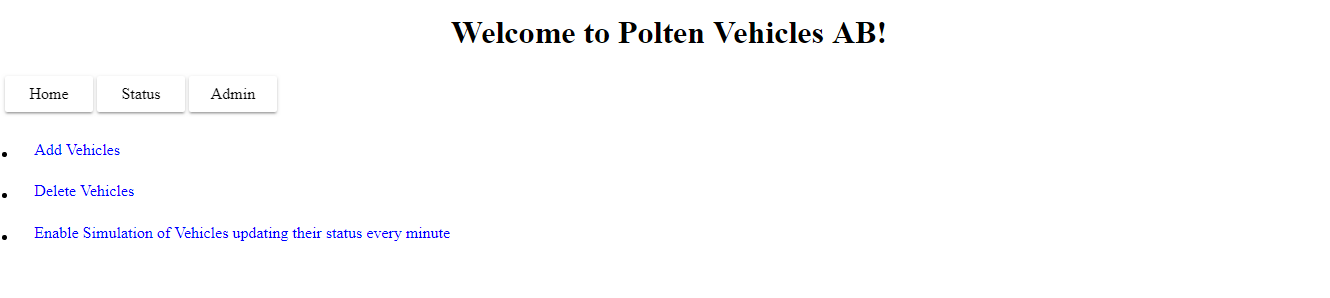


# Front end (localhost:4200/home)

Front page snaps **Home page**

****

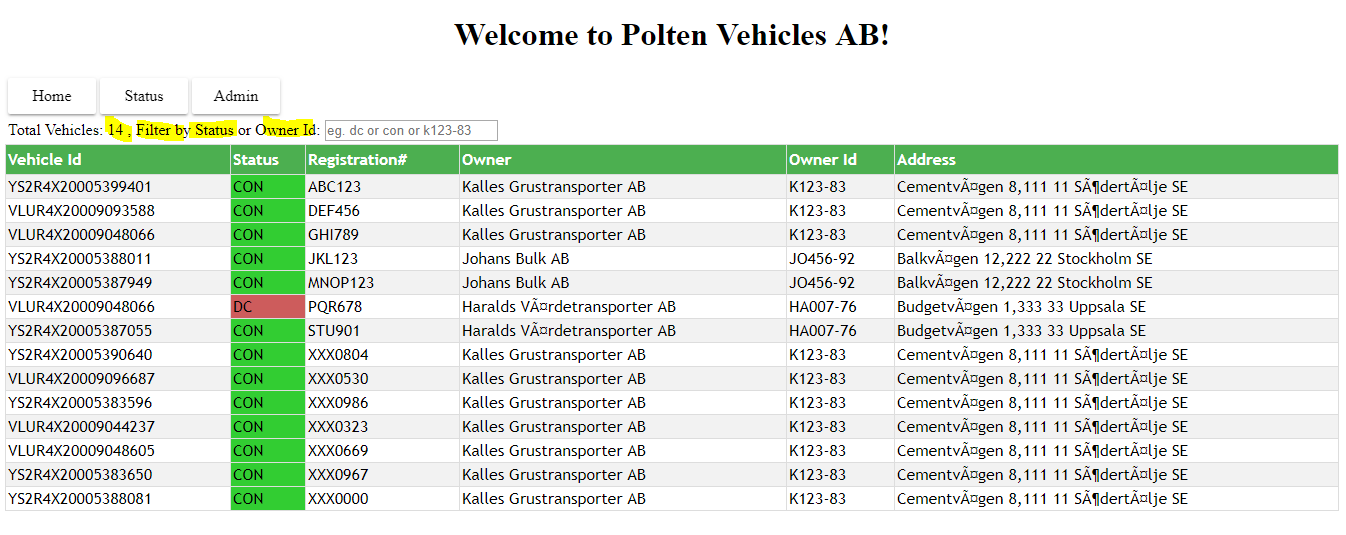
**Admin Page:**



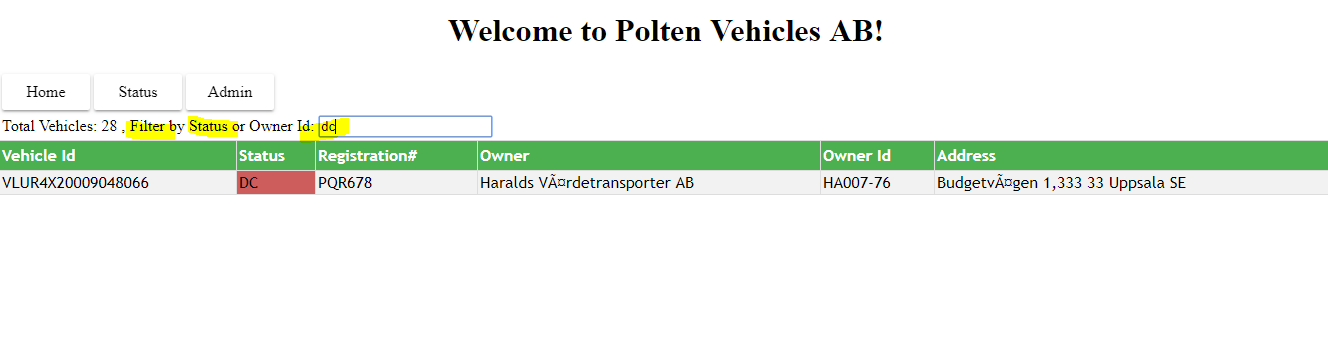
**Add some vehicles:**



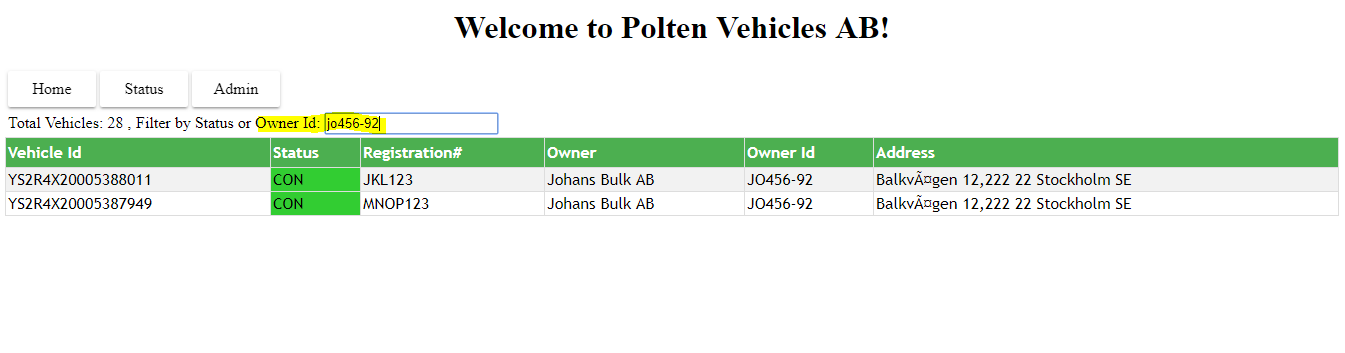
**Click Status Page:**



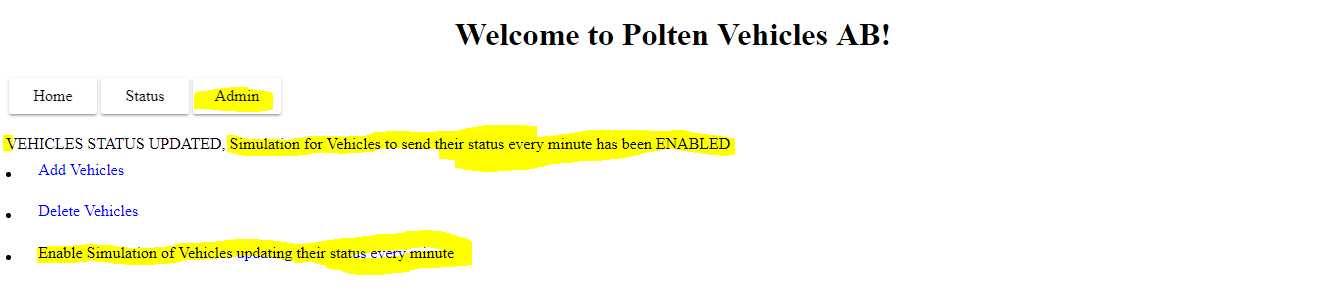
**Filter by Status: “dc”**



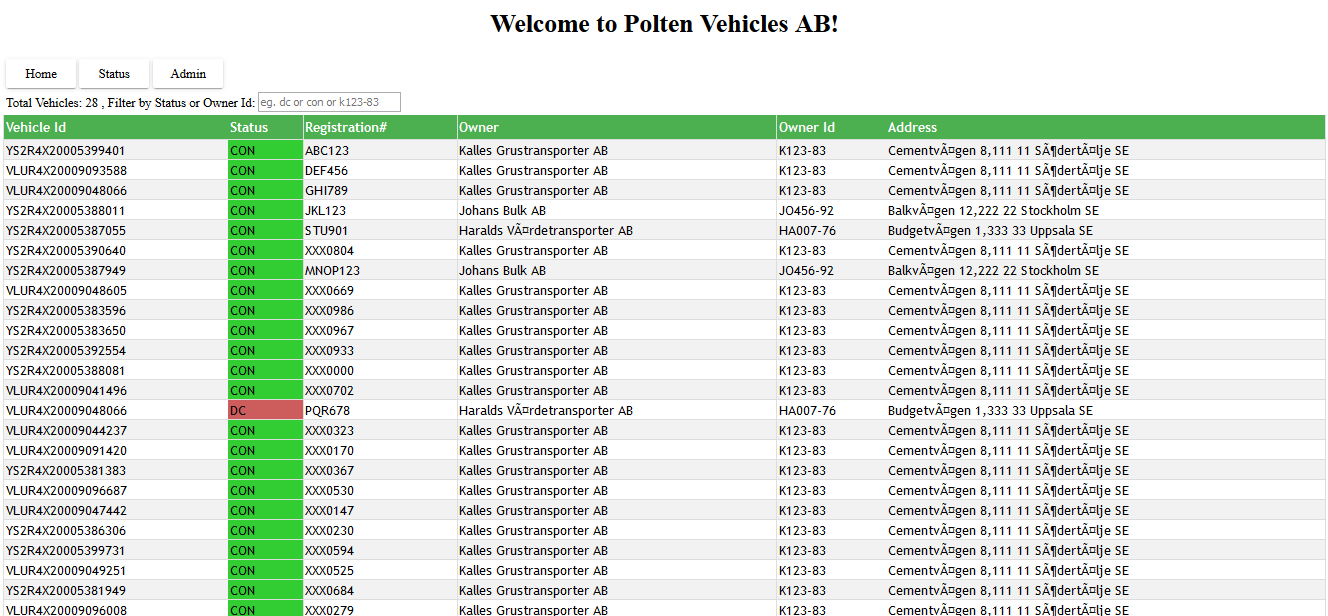
**Filter by Owner Id:**



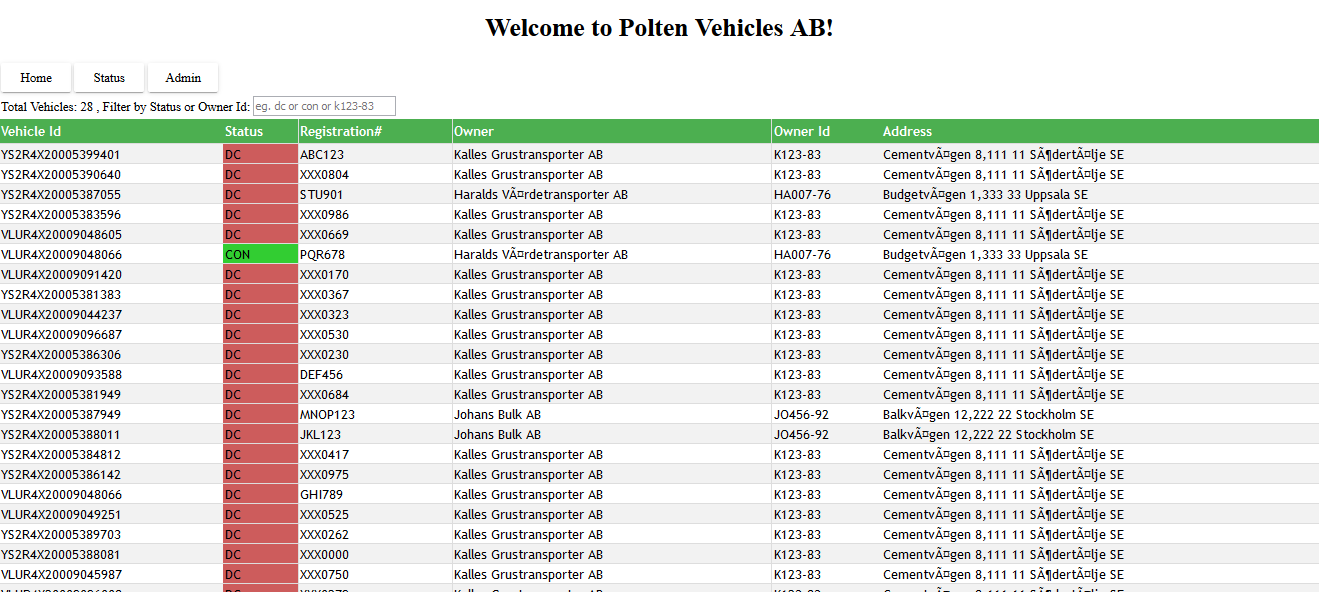
**Go To Admin page and Enable Simulation of vehicles sending their status every minute:**



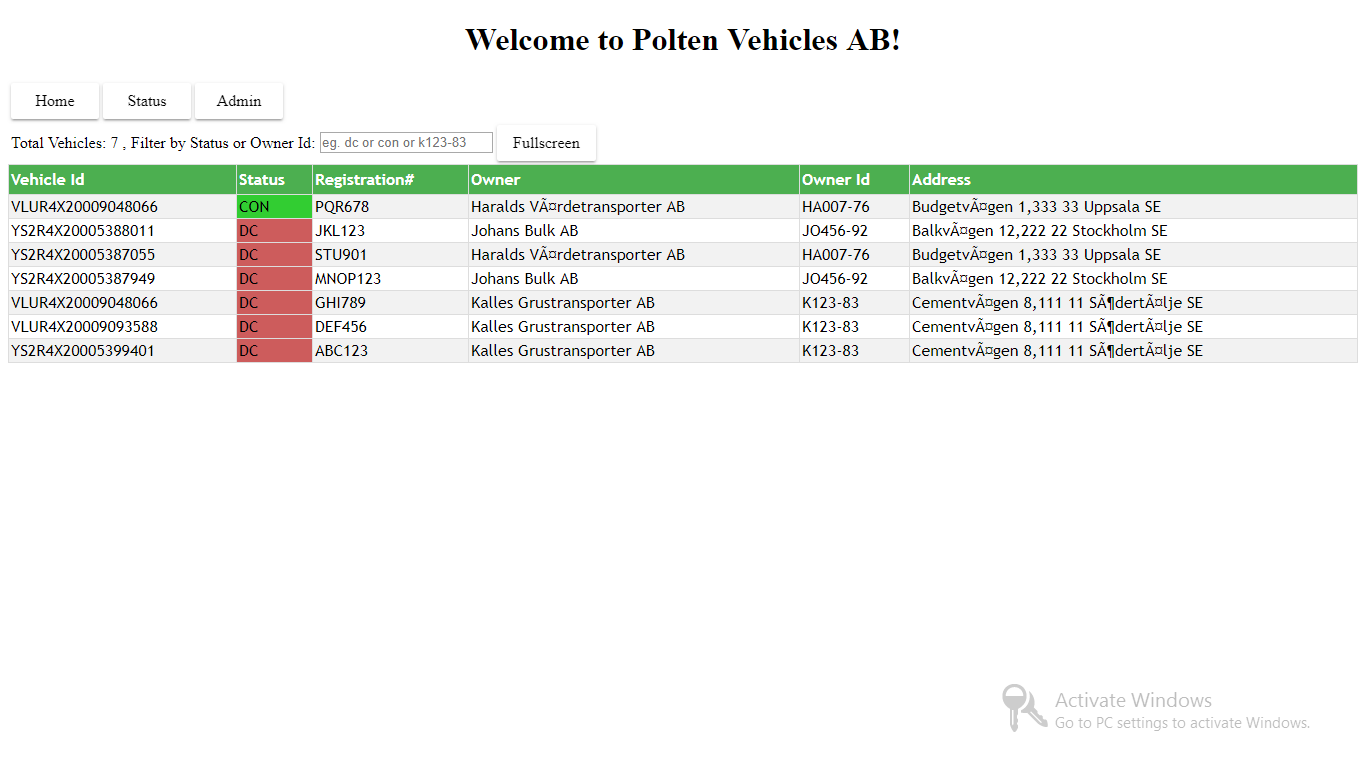
**Cobe back status page and leave the screen to track the status**



**After 1 minute:**



**Full screen functionality:**



# Assumptions

Vehicles are assumed to be **simulated** as request with current owner of vehicle & current status triggered from the angular components communicating service in interval of every 1 minute and status component refresh the current vehicle status into the UI. This setting can be enabled and disabled from the UI Screen in Admin section.

As there are no vehicle registration feature been requested. Some Test vehicles are added to test the applications functionality , there is a Admin section in the UI which can be used to control the same. Test can be added and deleted when needed. This is meant for only give an application working feeling not real vehicles are deleted when registered through API , There are some steps mentioned in the home page as well how to use the status application.