**Event driven car fuel consumption report**

**Components**

1. RabbitMQ running using docker
2. Event sender spring boot application
3. Event receiver spring boot application
4. Event trigger spring boot application using Thymeleaf UI.

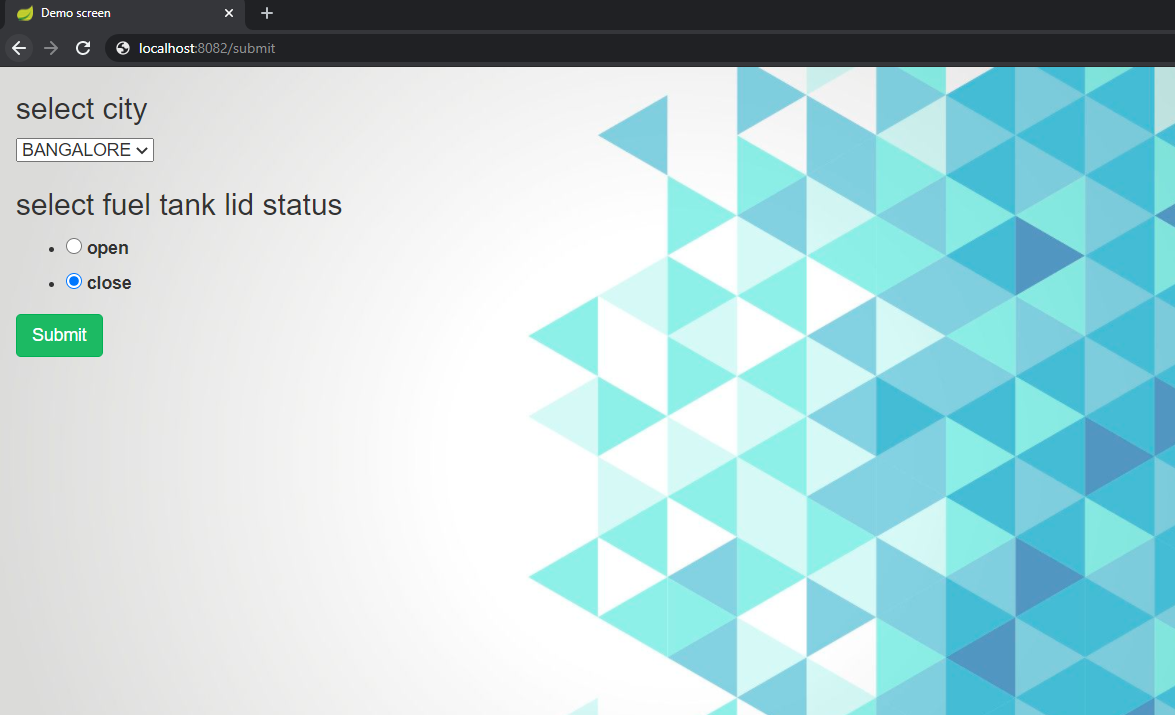
**System requirement**

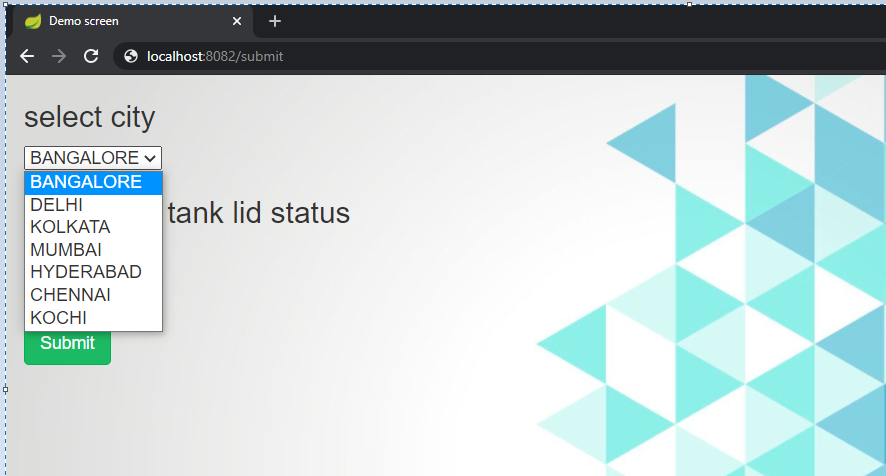
1. Docker
2. Gradle 5.0
3. JRE SE 1.8
4. Eclipse IDE

Source location – <https://github.com/shivananda207/ConnectedCarsPlatform.git>

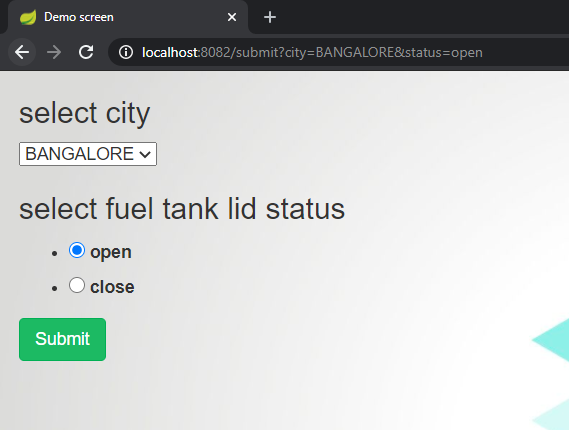
Screen

1. From UI screen city name can be select and by choosing fuel tank lid status event can be manually trigged

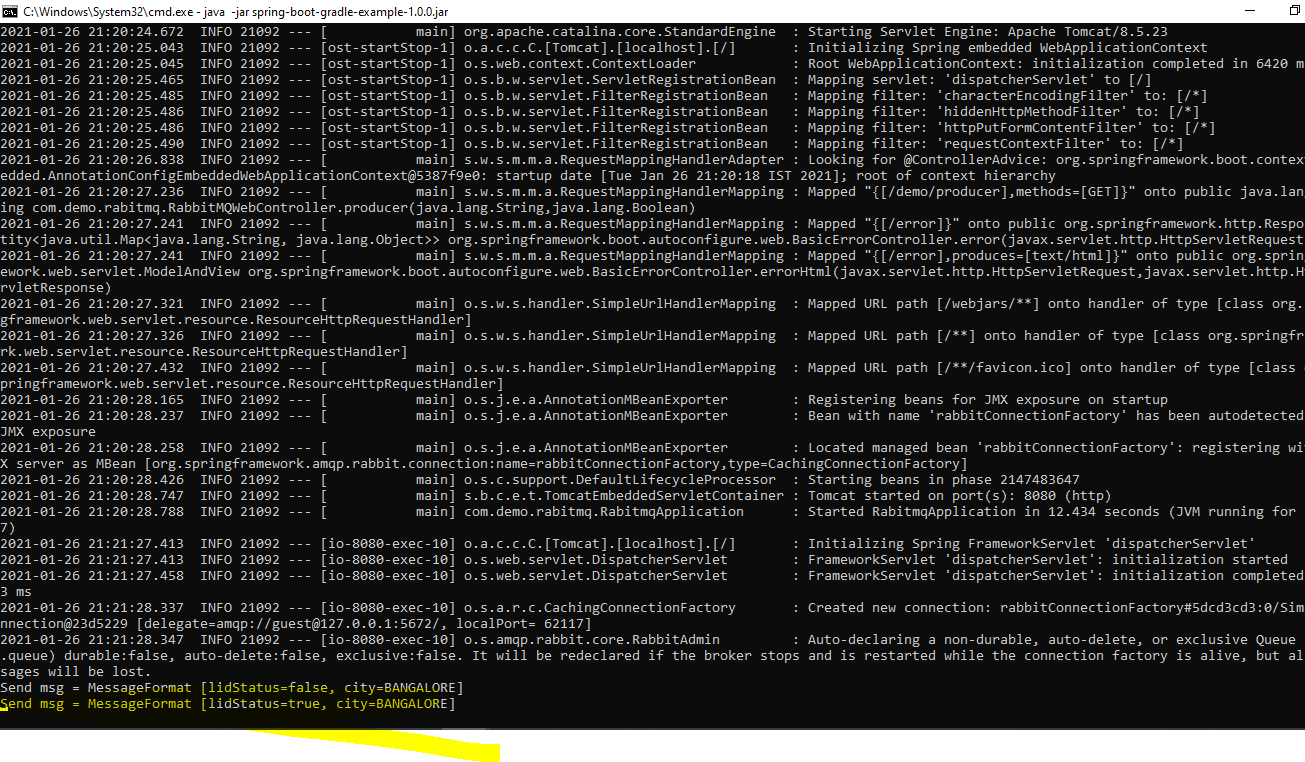




1. By choosing radio button event can be trigger manually

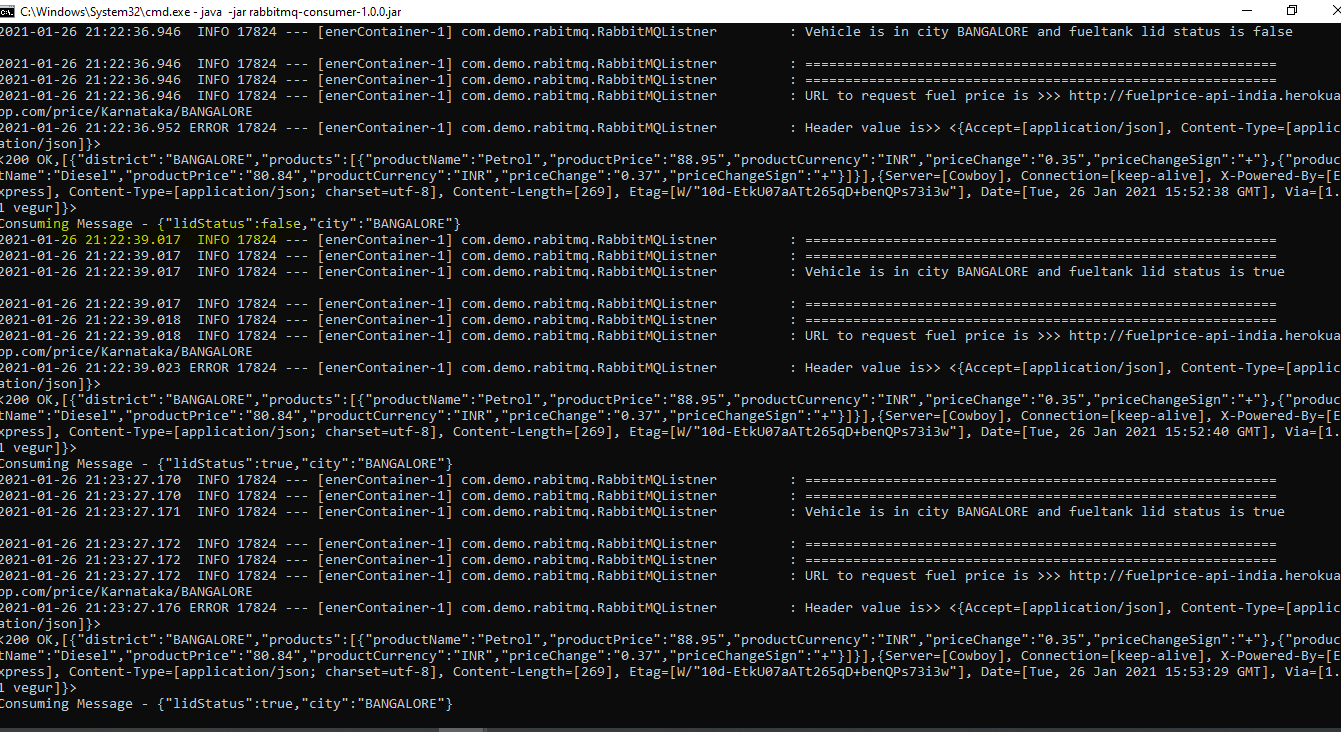


1. Event has Sent from RabbitMQ sender



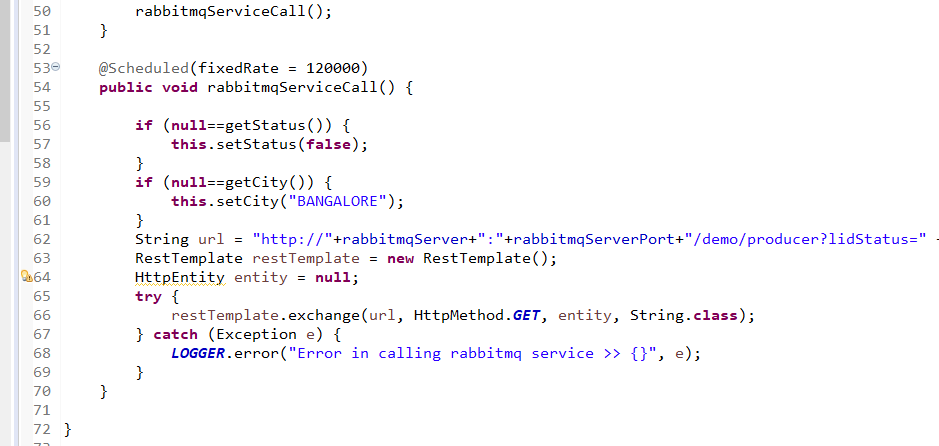
**Send msg = MessageFormat [lidStatus=true, city=BANGALORE]**

1. Event Consumer has received the message

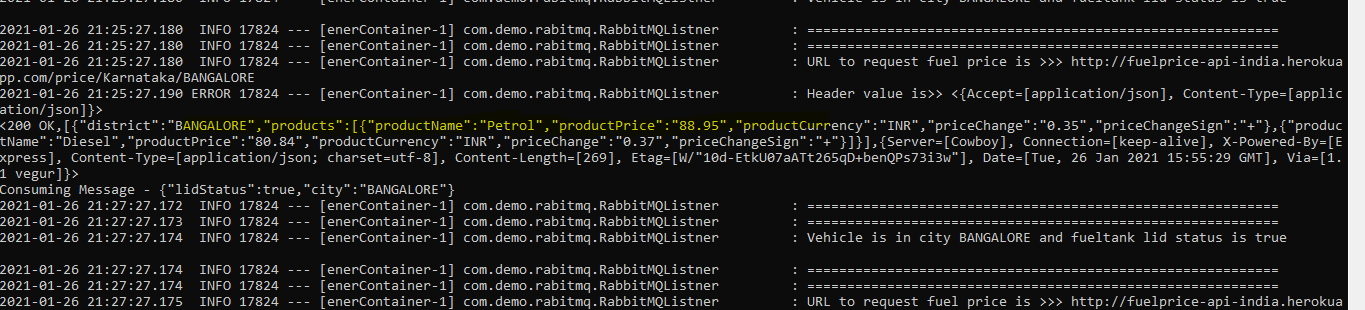


Event trigger client sends lid status every 120 sec

/eventtrigger/src/main/java/com/demo/eventtrigger/EventService.java



1. On receiving event, Consumer application makes API call to 3rd party fuel price report API and gets data based on city



Source code - /rabitmq-consumer/src/main/java/com/demo/rabitmq/RabbitMQListner.java

**To run application**

1. Pull rabbitmq docker image

Run rabbitmq

***docker run -it -d --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management***

1. Run rabitmq sender using jar (create using gradle 5.0)
2. Run rabitmq-consumer using jar (create using gradle 5.0)
3. Run eventtrigger using jar (created using gradlw 5.0)
4. Access UI running in <http://localhost:8082/submit>

**To do**

1. Parsing Fuel price from third party API with caching price for a day.
2. Calculating fuel lid open time interval.
3. Securing API using user right and APP ID in API request header.
4. Dockerizing the spring boot application.