Mitchell International Coding Challenge

Project Report

Vehicles Repository using Spring Boot

Developer:

Ankur Yadav

Research Associate

The University of Texas at Dallas

+1 (469) 258 8256

INDEX

Sr. No.	Content
1	Technology Stack
2	Features
3	Rest Services
4	In Memory Persistence
5	Automated Testing
6	Data Validation
7	Client

01. Technology Stack

In this project Spring Boot framework is used. The reason behind using Spring Boot Framework is this framework is built for building such micro services. Also, this framework provides autoconfiguration for difference dependencies and has built in features to develop web applications, rest controllers and testing.

The Complete Technology stack used in project is:

Framework	Spring Boot
In Memory Persistence	Spring Data JPA
Database	Apache Derby
Automated Testing	JUnit 5
Front End for Client	Material Bootstrap

02. Features:

According to the requirements, all the following features has been implemented in the project.

1	Rest Services for Vehicle Objects (GET, POST, PUT, DELETE)
2	Custom Querying on Objects (Using Make, Model or Year)
3	Data Validation on Parameters
4	In Memory Persistence
5	Automated Testing using JUnit 5
6	Client using Material Design Bootstrap

03. Rest Services:

Fully functional Rest Services are created for this project using Spring Boot Rest Controller. Following Methods are implemented to retrieve, add, update or delete data in the database.

Sr. no	URL	Type	Functionality
1	/vehicles	GET	Retrieve All vehicles' details in the database
2	/vehicles/{id}	GET	Retrieve Vehicle Details for the given ID
3	/vehicles/{param}/{value}	GET	Retrieve all vehicles' details with matching parameter and its value (filter)
4	/vehicles	POST	To add new Vehicle Object to Database
5	/vehicles/{id}	PUT	Update the vehicle object of given Id
6	/vehicles/{id}	DELETE	Delete the Vehicle Details for given Id

04. In Memory Persistence:

To Achieve this, we the Spring Data JPA is used. JPA stands for Java Persistence API. It is a specification that defines an API for object relational mappings and for mapping persistent objects.

In this project, Apache Derby database is used as it comes as built in with the framework. Each time the project is started, an instance of database is created, and that instance is used everywhere as a database in the project

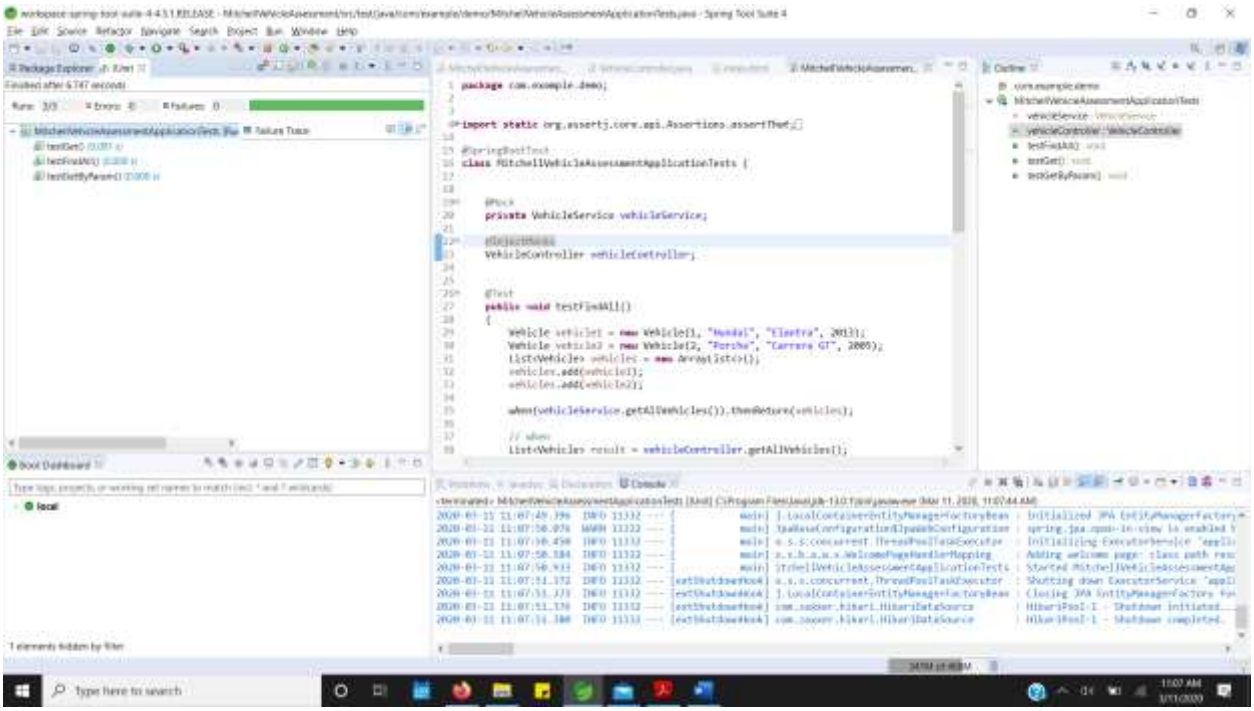
05. Automated Testing:

Java provides JUnit framework to perform automated testing on the Rest Services that we create. In this project JUnit 5 has been used to perform multiple tests.

Following three tests are carried out in the project.

1. Test to get all vehicle Objects
2. Test for get vehicle object by Id
3. Test for getting Vehicle by parameters

The following test result is observed after performing test.



All tests are conducted successfully.

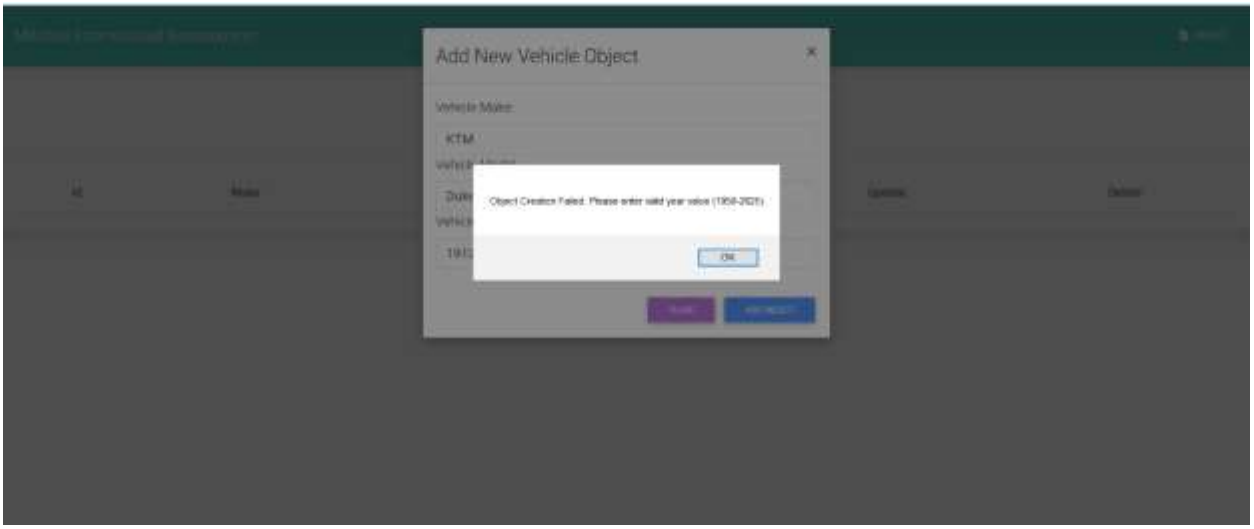
06. Data Validation:

The data validation is performed in the Rest Controller. Whenever any vehicle is added or updated following checks will be performed:

1. The Make value should not be null
2. The Model value should not be null
3. The Year value must be in between 1950 – 2050

If any one of these checks is failed then the Controller will return null indicating that data validation is failed.

Example:



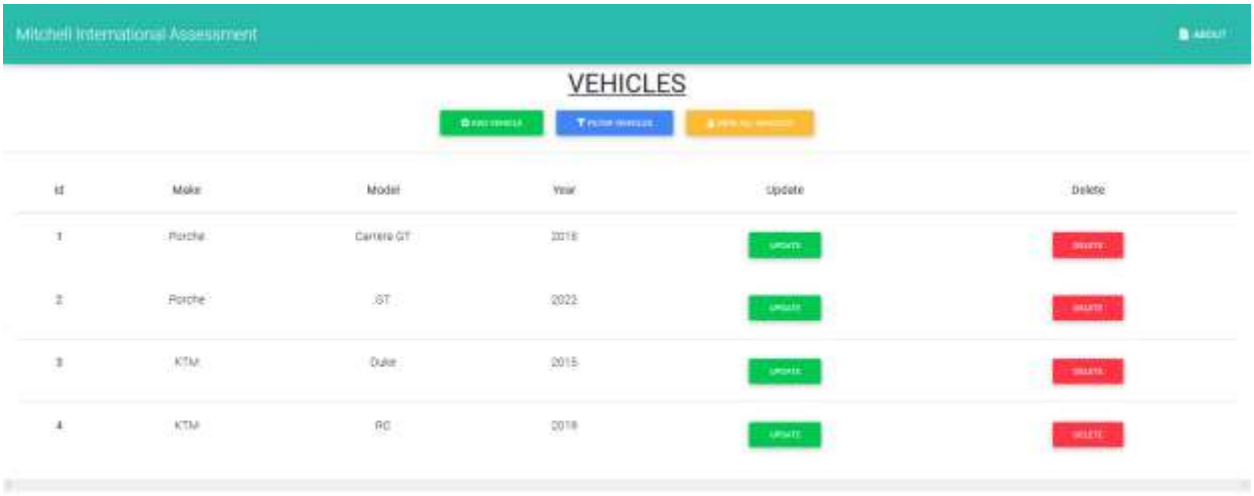
07. Client

A client with front end is designed using Material Design Bootstrap for proper interaction between Rest Services and external user.

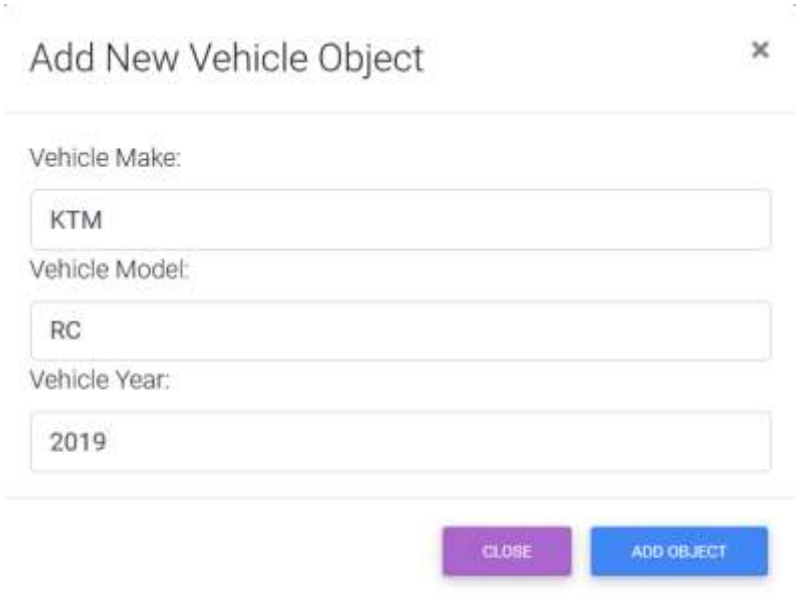
The client can be run on URL : *localhost:8080/*

The Client provides following features:

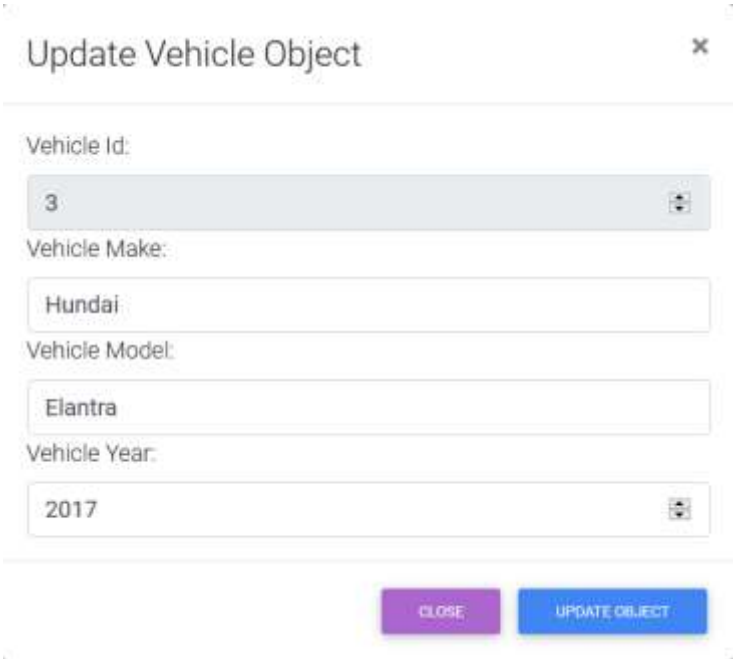
01. View All Vehicles:



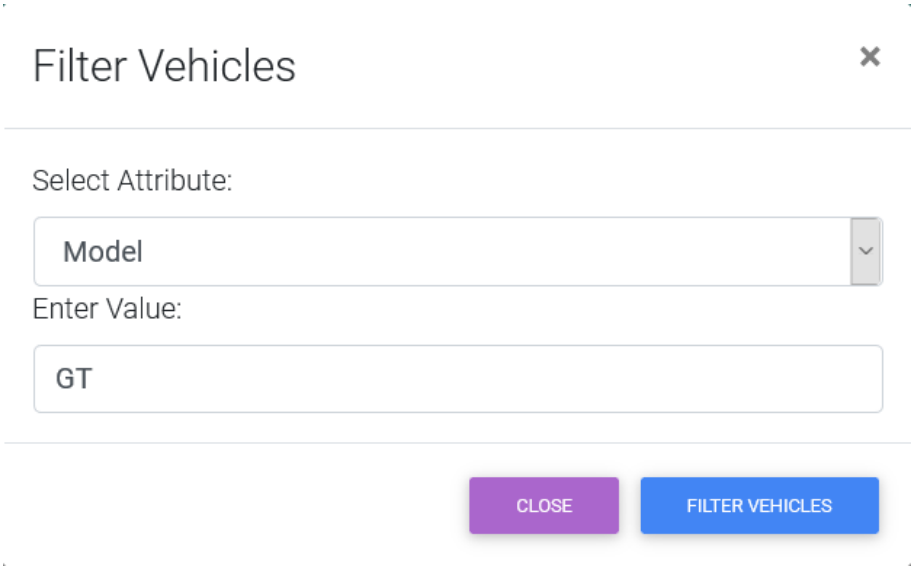
02. Add New Vehicle (Popup Modal)s:



03. Update Vehicle (Popup Modal):



04. Apply Filters (Popup Modal)



05. Delete Vehicle (Button)

(No Screenshot Available)