Location simulator application

This application is get coordinates (latitude and longitude) from origin and destination in specific interval. Implementation of this application done on JAVA and RestFul web services using spring boot framework. This application uses google map direction API to get polyline points. Service of this application can be called from browser or tool to get data.

Requirement gathering:

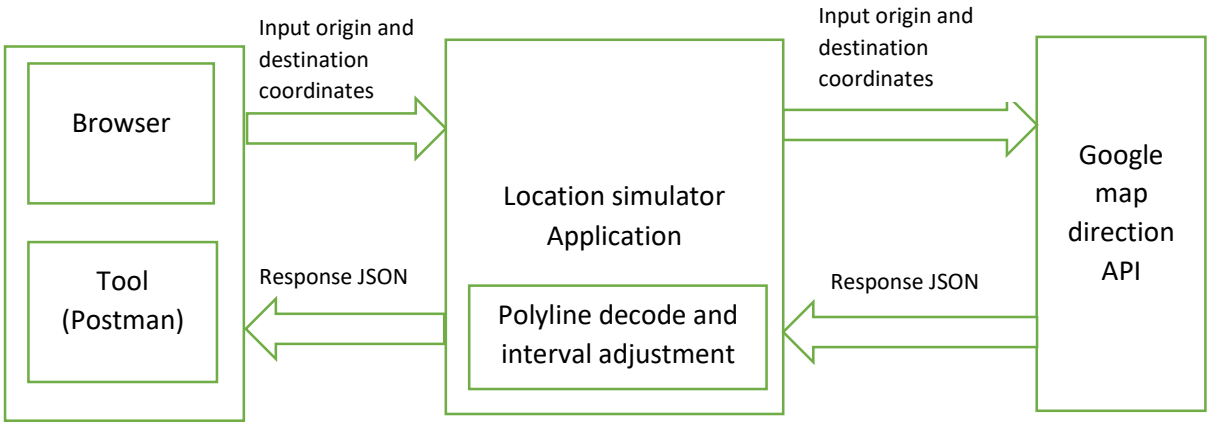
Create an API to get polyline coordinates in a specified interval between specified origin and destination.

Requirement analysis:

1. Get coordinates of line using origin and destination coordinates using google map direction API.
2. Process fetched polyline coordinates and restructure these coordinates to 50 meter interval between origin and destination.

Designing:

Following diagram shows how server configure internally and server-client communication.



Development:

Programming language: JAVA 8, spring boot

Build Tool: Maven

IDE: Spring boot tool suite 4

Application server: embedded apache tomcat server

Dependencies used in project:

1. spring-boot-starter-web: To develop Rest APIs and calling API using RestTemplete.
2. commons-math3: to perform mathematical operation like sin, cos and radian.
3. spring-boot-starter-actuator: for application monitoring
4. spring-boot-devtools: automatic server restart on detecting any changes.

Public API used:

Google's direction API used to get polyline paths between origin and destination.

REST endpoint of application:

http://{hostname}:{port}/locationsimulator/points/v0/origin/{origin}/destination/{destination}

Endpoint consists of:

1. Protocol: http
2. Hostname: on which server is deployed
3. Port: on which server is running
4. Origin: origin latitude and longitude points
5. Destination: destination latitude and longitude points

Example:

1. Request:
   URI:
   http://localhost:8080/locationsimulator/points/v0/origin/12.93175,77.62872/destination/12.92662,77.63696

   Header:

Header Name	Value
Accept	application/json
Content-Type	application/json

2. Response:

[
"12.93166, 77.62852",
"12.931251, 77.628711",
"12.930836, 77.628888",
"12.930429, 77.629083",
"12.930065, 77.629304",
"12.929931, 77.629744",
"12.929798, 77.630185",
"12.929688, 77.630632",
"12.929587, 77.631082",
"12.929502, 77.631535",
"12.929453, 77.631989",
"12.929566, 77.632436",
"12.929735, 77.632863",
"12.929865, 77.63329",

```
    "12.92976, 77.633732",
    "12.929545, 77.634137",
    "12.929354, 77.634555",
    "12.929143, 77.634962",
    "12.928931, 77.635369",
    "12.92872, 77.635776",
    "12.928506, 77.636182",
    "12.92831, 77.636597",
    "12.928078, 77.63699",
    "12.927749, 77.637304",
    "12.92743, 77.637391"
]
```