Take Home Outline
Blake Hudson

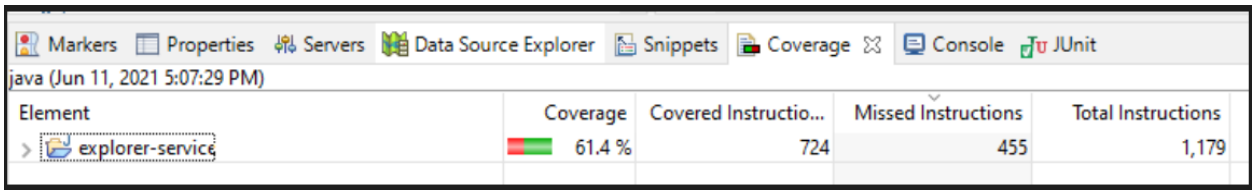
Scaling the project:

For scaling this project, if this was something deployed on AWS. I would assign users to be stored within DB's inside their availability zone. By analyzing their GPS coordinates, they can be sectioned off within regions by replicating the tables in each zone. With the possibility of either replica servers or a master server containing all of the data. This would result in querying only for stores within a region as well; which is more logical from the users perspective.

Docker Image:

https://hub.docker.com/r/hudsonbl360/takehome-explorer-service

Code Coverage: 61.4%



Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
explorer-service	61.4 %	724	455	1,179

Rest Compliance:

I would add HATEOS to the GET request for closest drivers to store. This would allow a paginated response if N is large.

```
ex: GET /api/drivers?StoreID=1234&N={100}
response:
  {
    drivers: [length = 10]
    nextPage: 2
    prevPage: 1
    currentPage: 1
  }
```

Endpoints:

1. End point should store users location
Kafka: id="DriverGroup" topics="topic-driverLocation" type=DriverLocation
POST /api/drivers/location

```
{
 "driverID": "m123@gmail.com",
```

```
 "latitude": 27.876,
 "longitude": -128.33
}
```

Response:

```
{
    "ok":"success saving data"
}
```

2. End point should store a stores location
Kafka: id="StoreGroup" topics="topic-storeLocation" type=StoreLocation
POST /api/stores/location

```
{
 "storeID": "1234",
 "latitude": 27.876,
 "longitude": -128.33
}
```

Response:

```
{
    "ok":"success saving data"
}
```

3. GET /api/drivers?StoreID={storeId}&N={#}
Kafka: id="GetDriverGroup" topics="topic-getDrivers" type=ArrayList<DriverLocation>

Response:

```
{
    "drivers": [
        {
            "dirverID":"m123@gmail.com",
            "distance":"100m"
        },
        {
            "dirverID":"m456@gmail.com",
            "distance":"200m"
        }
    ]
}
```

MySQL Tables:

Example Region Table Names:
- store_location_NA_west, store_location_NA_central, store_location_NA_east

Table Names:

- store_location

Table Content:

storeID (U)(K)	latitude	longitude
2231	27.124	-123.24
2250	27.124	-123.24

Example Region Table Names:
- driver_location_NA_west, driver_location_NA_central, driver_location_NA_east

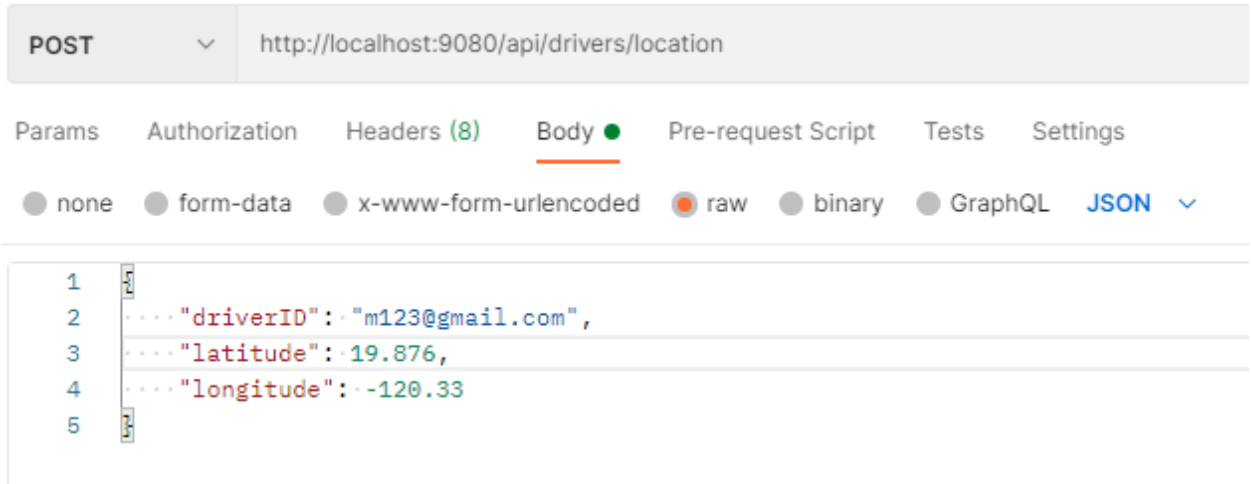
Table Names:
- driver_location

Table Content:

driverID (U)(K)	latitude	longitude
m123@gmail.com	27.124	-123.24
m456@gmail.com	27.124	-123.24

Testing with Postman:

Testing service should be able to take driver's current (latest)location.



Service should be able to take store configuration via REST API.

POST http://localhost:9080/api/stores/location ...

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "storeID": "1234",
3   "latitude": 27.876,
4   "longitude": -128.33
5 }
```

Service should expose a GET API to fetch N drivers around a store. StoreID and N should be taken as a query parameter to the API.

GET http://localhost:9080/api/drivers?StoreID=1234&N=2 ...

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
☑	StoreID	1234
☑	N	2
	Key	Value