COVID Tracker 0.1.0

We need to create an application to track COVID spread with the following aim:

- The application must provide APIs to upload data for specified locations.
- The application must provide search capabilities

Things to keep in mind

1. Time to finish this assignment is 2 days. It should not take more than 6 hours to finish the application.
2. Please don't post the solution on public Github, Bitbucket, Gitlab repositories.
3. In case you have any doubts feel free to email rahul.sharma@xebia.com for clarification.

Technology

We are expecting solution built using Java 8 and Spring Boot 2.x.

Evaluation criteria

Your code will be evaluated on following parameters:

1. Correctness
2. Code quality and cleanliness
3. Good use of OOPS concepts
    1. Extensible – easily adaptable to future enhancements
    2. Highly cohesive and loosely coupled
    3. Make use of design pattern wherever possible
4. Automation testing
5. REST API Design
6. Basic DevOps knowhow. We expect that you will submit `Dockerfile` with your project.
7. A `README.md` that clearly mentions steps needed to build and run the application.

Requirements

As part of the MVP, you only have to implement following user stories.

> You only have to build the REST API. Frontend is not in the scope of this assignment.

Story 1 : CSV Upload

Add an API to upload initial data in `CSV` format. The CSV will specify a `location` with the `number of people` in the following different stages of COVID. The stages of COVID are:

- Tested
- Confirmed
- Active
- Recovered
- Dead

On uploading the data the application must **replace** all previouss data for the specified cities.

Request :

```
Content-Type: text/csv

Id,Location,Tested,Confirmed,Active,Recovered,Dead
1,Delhi,3000,51,49,1,1
2,Gujrat,11000,545,544,0,1
3,Kerala,7000,145,144,1,0
```

Response : `201` with a JSON message.

> In case the file format is incorrect then following response should be returned with HTTP status code 400
>
> ```
> { msg : "File format not supported. Please use specified CSV format" }
> ```

Story 2 : Update CITY Data

Add an API to update data for a specific location. The request will contain `number of people` in different stages of COVID. All non specified stages can be assumed to be 0.

The application must hold data in an incremental manner. It will be used to determine analytics (growth, comparative analysis) across different stages of COVID

The request body is shown below.

```
{
  Tested:150,
  Confirmed:10,
  Active:1
 }
```

Response : `200` with a JSON message having the aggregated data for a location.

```
{
  Tested:3150,
  Confirmed:61,
  Active:50,
  Recovered:1,
  Dead:1
}
```

> The API must validate an existing Location in *case-insensitive* manner. An invocation with non-existing Location should return error in a valid JSON response message.

Story 3 : Search

The application should provide a search API which can handle the following filters :

Location (optional)

Selects the locations which are used to select a values. It is possile to select a single location or multiple locations. e.g. `location=Delhi` `location=Delhi&location=Gujrat`

The filter is **NOT** mandatory. In case no value is provided the application should serach with all available locations.

Type (optional)

This filter can be used to perform computation on the different stages of the selected data.

- total : computes the total of the different stages for selected data e.g. `type=total`
    - min : computes the minimum based on the selected column e.g. `type=min&selected=Active`
    - max : computes the maximu based on the selected column e.g. `type=min&selected=Confirmed`

The `type` operation return back an JSON list with single value.

```
[{
  Tested:3150,
  Confirmed:61,
  Active:50,
  Recovered:1,
  Dead:1
}]
```

The filter is **NOT** mandatory. In case no value is provided the application should return all selected values.