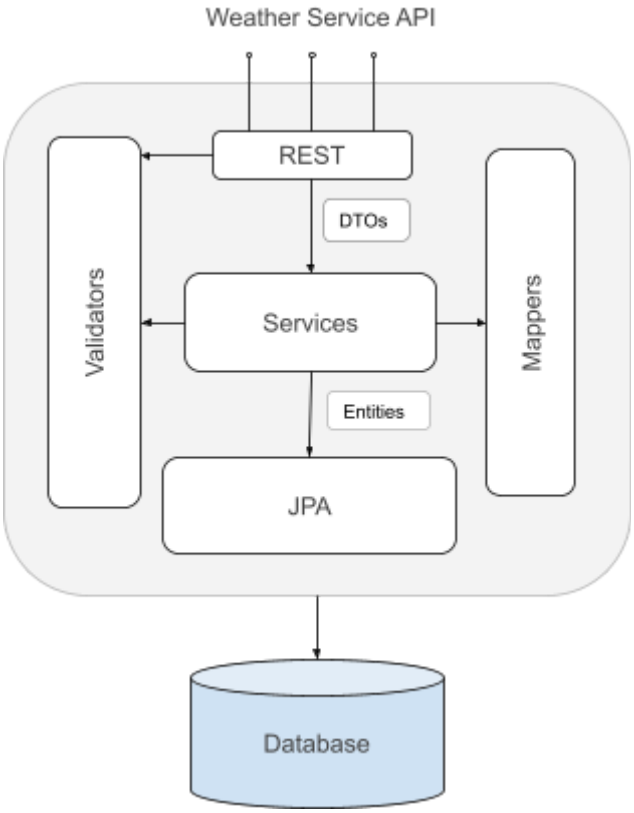
Weather Service

Summary

This document provides details of “weather web application”, intended to serve weather reports to its customers via REST endpoints. The REST endpoints provide the ability to add, query and delete weather data. The application has been developed using Spring Boot and JDK 1.8.

Design



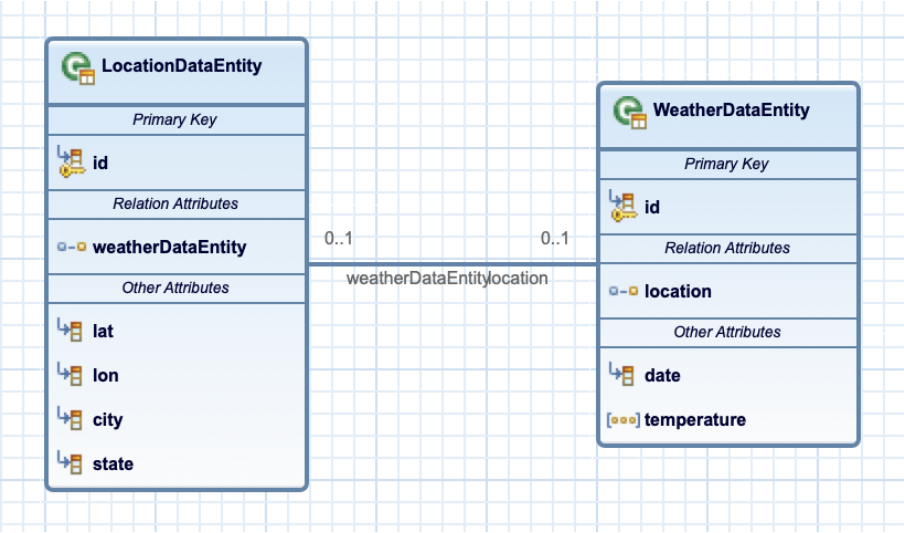
Tech Stack / Dependencies

REST	Spring controller
Persistence Layer	Spring JPA
Database	H2 Database
Validation	Apache Commons Validator
Unit Testing	JUnit
Logging	SLF4J
REST Client	Postman

REST Endpoints

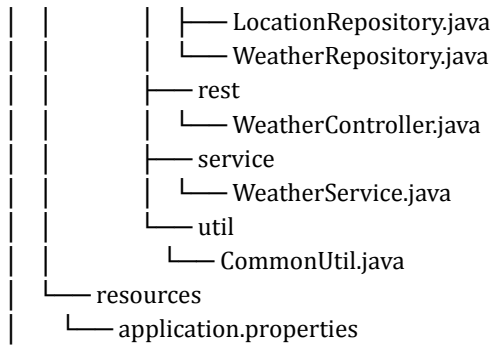
Method	Functionality	Endpoints	Response	Validations
POST	Adding new weather data	<host>/weather	201 OK Header: location = <host>/weather/{id}	400 Id already exixts Check for valid values ID, City, Date, Lat, Lon
GET	Returning all the weather data	<host>/weather	200 OK In ASC order of ID	
GET	Returning the weather data filtered by date	<host>/weather?date={date}	200 OK	Check for valid value of Date
DELETE	Erasing all the weather data	<host>/erase	200 OK	

ER Diagram



Folder Structure

```
.
├── pom.xml
└── src
    ├── main
    │   ├── java
    │   │   └── com
    │   │       └── travel
    │   │           └── weather
    │   │               ├── WeatherServiceApplication.java
    │   │               ├── dto
    │   │               │   ├── LocationDataDTO.java
    │   │               │   └── WeatherDataDTO.java
    │   │               ├── entity
    │   │               │   ├── LocationDataEntity.java
    │   │               │   └── WeatherDataEntity.java
    │   │               ├── repository
```



References

1. https://spring.io/guides/tutorials/rest/
2. https://commons.apache.org/proper/commons-validator/
3. http://modelmapper.org/getting-started/
4. https://www.baeldung.com/jpa-one-to-one