

Application Description:

The application receives weather data from sensors all around the world (5 sensors) that report the following metrics:

- Temperature
- Humidity
- Wind speed
- Pressure
- Cloud cover

The application allows users to get the current sensor readings from one or more sensors using the 'getSensor()' method in the API controller by specifying which city sensors to return in the path parameters. The application gets this information from a database. The application receives new sensor updates via the controllers 'updateSensor()' method in the API controller by specifying which city sensors to get updates for in the path parameters. The application allows user to query specific data by passing in path parameter arguments to produce specific endpoints based on the query. This is supported by the application's main methods. The application also allows users to get average readings for one or more metrics from one or more sensors which is again specified in the path parameters. The controllers 'getAvgMetric()' method is responsible for this functionality. The application allows users to filter data by date by specifying a date range in the path parameters. This is done via the controllers 'getBetweenDates()' method. This method is called within the 'getSensor()' and 'getAvgMetric()' methods for filtering results.

Technical Notes:

- The application is a rest-API
- Java and Spring Boot were used to build the project.
- The application is persisted using the MySQL database.
- Input handling and exception handling included
- Queried(run) using postman by passing in different parameterised URL's

How To Run:

- getSensor()
 - Acceptable URL formats
 - "/get/sensor/",
 - "/get/sensor/{sensorId}"

- `"/get/sensor/{sensorId}/{date1}/{date2}"`
 - `{sensorId}`, `{date1}` and `{date2}` are optional path variables. Legal values for the fields are as follows.
 - `{sensorId}` – "galwayIreland" or "munichGermany" or "texasUS" or "romeItaly" or "madridSpain" or "all" or "galway&munich&texas&rome" or "galway&munich&texas" or "galway&munich".
 - `{date1}/{date2}` – dates in the "dd-MM-yyyy" or "yyyy-MM-dd" format
- `updateSensor()`
 - Acceptable URL formats
 - `"/update/{sensor}"`
 - Path Variable `{sensor}` can have the following values:
 - "galway" or "madrid" or "munich" or "rome" or "texas"
- `getAvgMetric()`
 - Acceptable URL formats
 - `"/get/avg/{metric}"`
 - `"/get/avg/{metric}/{date1}"`
 - `"/get/avg/{metric}/{date1}/{date2}"`
 - Optional Path Variables `{date1}/{date2}` has same format previously specified (this is the format whenever required) and `{metric}` can have the following values.
 - "temp" or "windspeed" or "humidity" or "cloudcover" or any combination of the them.
- `getBetweenDates()`
 - Acceptable URL formats
 - `"/get/weather/{date1}/{date2}"`
 - `"/get/weather/{date1}"`