Hana Alizai
7/31/23
CS 361

Weather Microservice Communication Contract

1. **API Endpoint: The microservice will expose a RESTful API endpoint that accepts HTTP GET requests to fetch weather information.**

   Endpoint: /weather
   Method: GET

2. **Request Parameters: The client can specify the location for which weather data is requested using the following query parameters:**

   Query Parameters:
   - latitude (required): The latitude of the location.
   - longitude (required): The longitude of the location.

   Example API Request (call):

   GET http://your-microservice-url/weather?latitude=37.7749&longitude=-122.4194

   (If you are running the microservice locally on your development machine, the URL would be http://localhost:5000/weather. If you deploy the microservice to a server or a cloud platform, you need to replace localhost:5000 with the actual URL where the microservice is hosted.)

3. **Response Format: The microservice will respond with a JSON object containing weather information for the specified location.**

   Example Response:

```
{
  "location": "San Francisco, CA",
  "temperature": 22.5,
  "weather_description": "Partly Cloudy",
  "humidity": 60,
  "wind_speed": 10.5
}
```

Explanation of Response Fields:

- location: The name or description of the location for which the weather data is provided.
- temperature: The current temperature at the specified location (in Celsius or Fahrenheit).
- weather_description: A brief description of the current weather conditions.
- humidity: The current relative humidity percentage at the specified location.
- wind_speed: The current wind speed at the specified location (in km/h or mph).

4. **Error Handling: If the microservice encounters any errors during API requests or data processing, it will provide appropriate error codes and messages to inform the client about the issue.**

Example Error Response:

```
{
  "error": "Location not found",
  "status_code": 404
}
```

The status_code field will indicate the HTTP status code associated with the error, allowing the client to handle errors accordingly.

HTTP Status Codes:

200 OK: The request was successful, and weather information is provided in the response.
400 Bad Request: The request is missing required parameters/contains invalid data.
404 Not Found: The location specified in the request could not be found.
500 Internal Server Error: The microservice encountered an error during processing.

5. **UML Sequence Diagram:**

```
@startuml SequenceDiagram

actor Client
participant WeatherMicroservice
participant WeatherAPI

Client -> WeatherMicroservice: GET /weather?latitude=37.7749&longitude=-122.4194
WeatherMicroservice -> WeatherAPI: API Request
WeatherAPI --> WeatherMicroservice: Weather Data (JSON)
WeatherMicroservice --> Client: Weather Data (JSON)
```

```
@enduml
```

- The Client sends a GET request to the WeatherMicroservice with the latitude and longitude query parameters.
- The WeatherMicroservice makes an API request to the WeatherAPI, passing the provided location coordinates.
- The WeatherAPI processes the request and returns weather data in JSON format to the WeatherMicroservice.
- The WeatherMicroservice receives the weather data from the WeatherAPI and sends it back as a JSON response to the Client.