

API ARCHITECTURE

(Weather app)

1. Microservice architecture
2. API gateway
3. Weather data service
4. Cache service
5. Monitoring and Logging

Microservice architecture

For greater flexibility , scalability and ease of use microservice architecture can be used. The system is split into small services .

Services

1. Fetching weather data
2. Managing location

Communication

HTTP can be used for the communication purpose . Since it is lightweight and easy to use

Load balancing

Load balancing can be used to distribute incoming requests across multiple instances of a service, improving the performance and scalability of the system

API gateway

Is used to get requests from the front-end.

Scope: Data of Multiple cities

Api which returns weather data of multiple cities .

Api also filters the result by city name or city code.

Api should return the data in pagination

End point: /weather-multiple-cities?filterType=\$type&limit=\$num

\$type: filter type (code/name)

\$num : pagination limit

Method: POST

Request:

Field	Type
cityCodes	[city codes/City name]

Response

Field	Details
data	Return data
message	string
error	boolean
status	Status code

Scope: Detailed Forecast for the next X days.

X days should be decided by the user.Data should be very detailed.

End point: /weather-forecast/cityCode\$cityCode?noOfDays=\$noOfDays

\$cityCode: city code to identify city

\$noOfDays : no of days of forecast needed

Method:GET

Scope: Filter the data by any particular city, any particular date , any particular moment

End point: /weather-data-by-city/\$cityCode?date=\$date&time=\$time

\$cityCode: city code to identify city

\$date : date

\$time: time

Method:GET

Scope: Current weather conditions of any particular city

End point: /weather-data-by-city/\$cityCode

\$cityCode: city code to identify city

Method:GET

Weather data service

openweathermap: Weather forecasts, nowcasts and history in a fast and elegant way

Cache service

Redis: Redis is an in-memory data structure store that can be used as a database, cache, and message broker. It is fast and scalable, making it a popular choice for weather apps that need to store and retrieve large amounts of data quickly.

Monitoring and Logging

Application Performance Monitoring (APM): APM tools can be used to monitor the performance of your weather app, including response time, memory usage, and resource utilization. This information can be used to identify performance bottlenecks and optimize the app for better performance.

Error Monitoring: Error monitoring tools can be used to automatically detect and notify you of any errors or exceptions that occur in your weather app. This allows you to quickly address and resolve any issues that may arise.