**Bioquix Backend Assignment – API Proxy with Heartbeat Monitoring**

This project implements a resilient, Dockerized backend system that proxies weather APIs and monitors their health using a separate Heartbeat service.

Built using ‘Spring Boot’, this architecture demonstrates fault tolerance, retry logic, inter-service communication, and container orchestration using Docker Compose.

**Features Implemented**

Two Spring Boot services ----- API Proxy + Heartbeat

Dockerized microservices ------ Both services run as containers

Health monitoring ------ Heartbeat pings `/weather/health` every 5 seconds |

API switching with fallback ------ If API1 fails, switches to API2

Retry on timeout ------ Retries once on connection timeout

Stub fallback ------ If both APIs fail, returns a stub response

Real-time status feedback ------ API Proxy sends POST status to Heartbeat

No database dependency ------ Pure REST-based communication

Docker Compose integration ------ Easy orchestration with `docker-compose up --build`

**Setup Instructions:**

**Prerequisites:**

Ensure the following are installed:

- Java 17

- Maven

- Docker

- Docker Compose

**Verify by running:**

bash

java -version

mvn -v

docker -v

docker-compose -v

**Project Structure:**

bioquix-assignment/

├── docker-compose.yml

├── api-proxy/

│ ├── Dockerfile

│ ├── pom.xml

│ └── target/api-proxy.jar

└── heartbeat/

├── Dockerfile

├── pom.xml

└── target/heartbeat.jar

**Build Instructions:**

1. Build the services

cd api-proxy

mvn clean package -DskipTests

cd ../heartbeat

mvn clean package -DskipTests

**2. Rename .jar files**

If the files are named like:

api-proxy-0.0.1-SNAPSHOT.jar

heartbeat-0.0.1-SNAPSHOT.jar

**Rename them to:**

api-proxy.jar

heartbeat.jaror update your Dockerfiles accordingly.

Run with Docker Compose

From the bioquix-assignment/ root folder:

docker-compose up --build

**Testing:**

http://localhost:8080/weather?city=London

Check health endpoint:

http://localhost:8080/weather/health

Check logs (heartbeat container):

docker logs -f heartbeat

output:

[HEARTBEAT OK] API Proxy is alive

[STATUS RECEIVED FROM API PROXY] Using API1

**Architecture Explanation:**

Inter-Service Communication

**+--------------------------+ +---------------------------+**

**| | | |**

**| HEARTBEAT SERVICE | | API PROXY SERVICE |**

**| | | |**

**| 1. GET /weather/health | ---------------------> | Health check every 5 sec |**

**| | | |**

**| 2. POST /status | <--------------------- | Sends status info |**

**| Receives status updates | | (fallbacks, errors) |**

**| | | |**

**+--------------------------+ +---------------------------+**

Heartbeat uses @Scheduled to ping /weather/health every 5 seconds.

API Proxy sends real-time updates via POST /status to Heartbeat.

Communication is RESTful and containerized using Docker networking.

project Developed By: **P.GOPICHAND**

📅Date: 28-May - 2025