Hawaiian pizza

How to run locally

The application requires docker to be installed locally (or to install all the dependand services manually). Start up the docker-compose with the file in the root and the app is ready to run.

Sample HTTP requests

There are sample HTTP requests in the root of the project in the `sampleRequests.http` file. The requests contain correct auth headers. They can be ran from the IntelliJ Idea Http client (only part of the paid version).

Sample data

Default users and pizzas are created using SQL load script. Other data should be created using sample HTTP request

Authentication

Authentication is based on Basic Auth, so with every request to a protected endpoint, username and password must be added as headers.

Application Review

PART 1

1. Architecture Assessment Summary
    - The overall architecture of this application is inconsistent and vulnerable.
2. The top 4 most valuable improvements required are:
    - API first approach
        - This allows a contract definition drive the application implementation. Taking an API first approach will improve functional understanding and integration of the services offered by the application to consumers. An API first approach also implies having a contract

specification that will enable multiple stakeholders collaborate on evolving this application.
    - Layering and avoiding leaky abstraction
        - The implementation does not follow a consistent layering approach.
        - The service layer should abstract the persistence layer and use data transfer objects that do not expose internal application data.
    - Database change management
        - Database definition and manipulation should be versioned and managed outside the application logic.
    - Application logging.
        - Logs needs to be generated and captured in order to monitor and provide insight into the application at runtime.
3. Minimum requirement for production readiness
    - The application should have an API and implement semantic versioning to provide information about the deployed API version at runtime.
    - Database change managment must be enforced and version migration information should be available at runtime.
    - The application must provide logging and metrics for monitoring and health checks.
    - The application should have resilience baked so that it can handle high user traffic and intermittent failure of backing services eg database access failure. The top 2 resilience implementation required are:
        - Circuit breakers and retry mechanisms
        - Pagination implementation at the Rest layer and persistence layer.
    - Identitiy and access management concerns should be improved and delegated to an external application.
    - The docker script should be improved so that the application process is run by user with restricted privileges.
4. The following tools and and frameworks would be benefical to the applicaion:
    - Logback - for application logging
    - Spring boot actuator - for gathering application metrics and monitoring application health.
    - Liquibase or Flyway for database change management and migration.
    - Prometheus and Grafana or ELK stack - for collecting and visualising application log data.
    - Hysterix - for configuring circuit breakers

PART 2

The new rule has been implemented based assumptions that have not been explicitly clarified by the client:

Assumptions:

- That the new rule will apply exclusively to an order quantity of 3
- That the new rule will be applied at a higher order of precedence over existing rules.
- That the new rule may override subsequent rules.