Alten Challenge for Hotel
Post-Covid Scenario

Release: 1

Alten Challenge for Hotel Post-Covid Scenario Date this document was generated: 2 September 2019

Modification History

Release	Comments
Release 1	Backend application containing the requirements specified in the statement.

Overview

Implementation a booking API for the very last hotel in Cancun assuming the scenario of Post-Covid situation. People are now free to travel everywhere but because of the pandemic, a lot of hotels went bankrupt. Some former famous travel places are left with only one hotel. Taking in count the following requirements:

- API will be maintained by the hotel's IT department.
- As it's the very last hotel, the quality of service must be 99.99 to 100% => no downtime
- For the purpose of the test, we assume the hotel has only one room available
- To give a chance to everyone to book the room, the stay can't be longer than 3 days and can't be reserved more than 30 days in advance.
- All reservations start at least the next day of booking,
- To simplify the use case, a "DAY' in the hotel room starts from 00:00 to 23:59:59.
- Every end-user can check the room availability, place a reservation, cancel it or modify it.
- To simplify the API is insecure.

Technical Documentation

Dependencies and Tools

This section listed and describe the different tools, technologies, frameworks (dependencies) that are used in this software implementation

NAME	DETAILS
Spring Boot	Enterprise framework that offers mostly auto configuration for Spring Framework's modules used for this application.
Spring Boot Starter Web	Spring boot module that allows web development supporting auto configuration in Java leveraging the Spring Boot capabilities. Starter of Spring web uses Spring MVC, REST and Tomcat as a default embedded server.
Spring Data JPA	Spring Data is a part of the Spring Framework and its goal is to significantly reduce the amount of boilerplate code required to implement data access layers for various persistence stores since it adds an extra layer of abstraction on the top of our JPA provider. The JPA Provider for this implementation is Hibernate.
Lombok	Library that offers a several Java annotations to avoid repetitive and bowler plate code, being capable of creating necessary code for most of the Java classes by just decorating the class with the annotations supported by the library to declare: Getter, Setters, Constructors, ToString, equals, among others.
Postgresql	Database Management System used for this implementation where the data is going to live in. Version: latest.
Liquibase	Liquibase Community is an open source project that helps millions of developers rapidly track, version, and deploy database schema changes. Enables Java applications to apply Databases migration.
SpringDoc OpenApi	library helps automating the generation of API documentation using spring boot projects.

Docker	It's an open source project that offers a technology to create and use containers. A container is a standard unit of software that packages up code and all its dependencies to allow applications run quickly and reliably from one computing environment to another.
Jacoco	Software metric used to measure how many lines of our code are executed during automated tests (Code Coverage Reports)
Angular	Angular is an open source UI framework. In this project angular was used to build a friendly User Interface to interact with the BE services. The Angular version running on this project is 12.1.2
Angular Material	It's a UI component library that leverages angular capabilities and offers nice and friendly web components for angular based on Material Design. The Angular material version used on this project is 1.2.11
Bootstrap	It's a styling library for Web Development in HTML offering great alternatives and solutions for CSS styling. The bootstrap version used on this project is 5.1.3

Design Patterns

This section lists the Design Patterns uses when this implementation was being developed

Name	**Description**
MVC	Used to segregate the business logic of the application in mainly 3 Layers: Model, View and Controller.
Repository	Levering Spring Data Repositories implementation this application implicit uses the Repository pattern which provide a mechanism for encapsulating storage, retrieval and search behavior that emulates a collection of objects

DTO	A Data transfer object is an object that encapsulates specific data according to the transaction and sends it from one subsystem of an application to another.
Singleton	Singleton is the default scope for Spring beans, in the application several Spring beans are created using the default scope, so implicitly this pattern is used by the application. The main goal of singleton is to only have 1 instance of that object across the application
Builder	Levering Lombok capabilities, the builder pattern was used to create new object instances.

Prerequisites

This section lists the prerequisites to consider to be able to run the application

Item	**Description**
1	Docker Suite needs to be installed in the host machine where this project is going to be run.
2	The application is going to use some of the host machine ports. Be aware that ports 8080, 5432 and 4200 need to be available (not used) by the moment you run the application.
3	You need to be connected to the internet since docker is going to download project dependencies to subsequently be able to start up the app.

Run Steps

This section lists the run steps to follow to be able to run the application

After all the prerequisites are guaranteed, you can proceed to run the project. for that, you'll need to open a terminal and navigate to the project root folder. Once you're there, you just need to run the following command:

```
docker-composer up
```

Docker will start to build up the images, the first time run may take some minutes since some dependencies are going to be downloaded to finally build and run the project. When you get prompted with the Spring Boot messages confirming the project is running on the 8080 port. You can open up th
e sample postman collection provided (postman-collection folder). You're good to go and interact with the application.

If you want to access the UI to interact with the API, open the following link in your preferred browsers: http://localhost:4200

For the API documentation, you can access the Swagger-UI docs interface by accessing: http://localhost: http://localhost:8080/api/swagger-ui. If you're interested in further technical documentation about the methods and classes definitions, you can go to the "Documentation Folder > javaDocs " and open the index.html file

PD: If you're interested in watching the latest code coverage report, you can go to "Documentation Folder" and open the index file in the "Jacoco Report" folder. In case you want to generate the report yourself just. Perform Maven clean install as the first step then you can execute mvn jacoco:report command. Once the command is executed, navigate to /target/site/jacoco and open up the index.html file with you prefered browser

Once you're done interacting with the application and you want this to shut down, go to the same terminal you executed the comment to start it up, press Ctrl + C and type in the following command:

```
docker-composer down
```