

Problem Statement: Build a Hotel Search application.

Requirement: (Rest Apis)

1. Insert Hotels into the system. Single and in bulk.
2. Get all the hotels in the system.
3. Get hotels based on a pattern i.e. Starts with/Contains for the hotel name.
4. Update the hotel information. Single and in bulk.
5. Delete hotels from the system.
6. Include User authentication with different roles i.e. Admin and User. Admins should only insert, update and delete hotels. Users can only search for the hotel.
7. The Hotel information should be comprehensive with the availability of the rooms, cost of the rooms and the amenities the hotel has.
8. Junit Test cases

Tech Stack

Mandatory:

1. Java
2. Spring boot2
3. Datastore like MySql, Cassandra, CouchBase etc.
4. Git Bash
5. Maven

Optional:

1. Ides like Eclipse or IntelliJ Idea.
2. Git clients like Tortoise Git or Git Extensions
3. Rest Clients like PostMan

Objectives:

1. Follow proper java naming conventions.
2. Design the application in a layered manner.
3. Use Git and commit to it with proper commit messages. Follow <http://karma-runner.github.io/4.0/dev/git-commit-msg.html>. Commit to local repository is enough and there is no need to commit to the remote repository. (test app provided).
4. Proper handling of error scenarios.
5. Provide proper documentation to set up the system (application and database) and run the application. Document the rest APIs [Url Pattern, Http Methods, Http Code (Success and Error), Request and Response (Success and Error)] so that we run the application and test it.

Test App Details:

1. A hotelsearch.zip is provided which is a starter app for the test. A Hello controller is provided to get an idea about the Rest apis.
2. Import to the ide, Run the class HotelsearchApplication as a normal java application or from the command line run “mvn spring-boot:run” in the project folder. Use a Rest client for eg:- PostMan to run the Rest Apis <http://localhost:8080/hello>.
3. A local git repository is initialized with a initial commits with proper commit messages. All commits should follow the commit message format.