

# TEST PLAN FOR INVENTORY-API

## *ChangeLog*

Version	Change Date	By	Description
1.0	28-06-21	Sarthak Verma	API testing

<b>1</b>	<b>INTRODUCTION .....</b>	<b>2</b>
1.1	SCOPE .....	3
1.2	QUALITY OBJECTIVE .....	3
1.3	ROLES AND RESPONSIBILITIES .....	4
<b>2</b>	<b>TEST CASES .....</b>	<b>4</b>
2.1	ENDPOINTS.....	4
2.2	TEST CASES .....	4
2.3	TEST COMPLETENESS.....	5
<b>3</b>	<b>TEST DELIVERABLES.....</b>	<b>5</b>
<b>4</b>	<b>RESOURCE &amp; ENVIRONMENT NEEDS.....</b>	<b>6</b>
4.1	TESTING TOOLS .....	6
4.2	TEST ENVIRONMENT.....	6
4.3	AUTOMATION .....	6
<b>5</b>	<b>TERMS/ACRONYMS .....</b>	<b>6</b>

# 1 Introduction

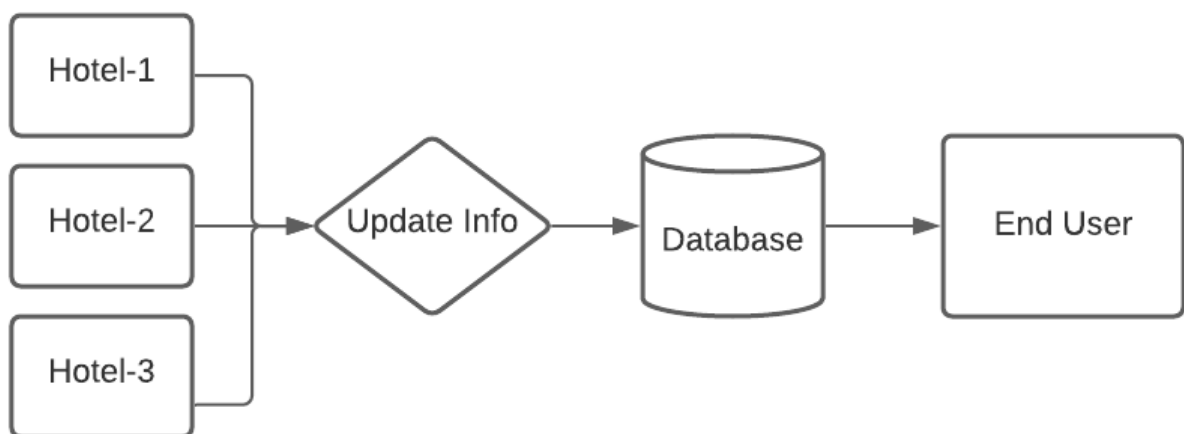
Inventory API application is a simple restassured application used for fetching/inserting the hotel inventory data. The Application includes fetching data like price per night and rooms remaining for a specific date in a hotel. The application also includes functionality for hotels to update price per night and remaining rooms for a specific date.

The Application also contains information about hotels like hotel name, city, country and types of rooms available in the hotel.

The Application is used by multiple clients, some front-end web application for hotel owners to manage their hotel inventory, calling these endpoints via some FE-Client library and some booking-api calling our Fetch Inventory endpoint to check availability for any hotel searches customers has chosen on the website, like [www.agoda.com](http://www.agoda.com)

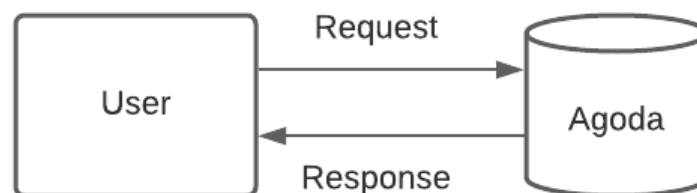
## Basic Flow-

Hotels will use update API for inserting/updating data in the database. End users/clients would use the fetch data API for getting the updated data from the DB.



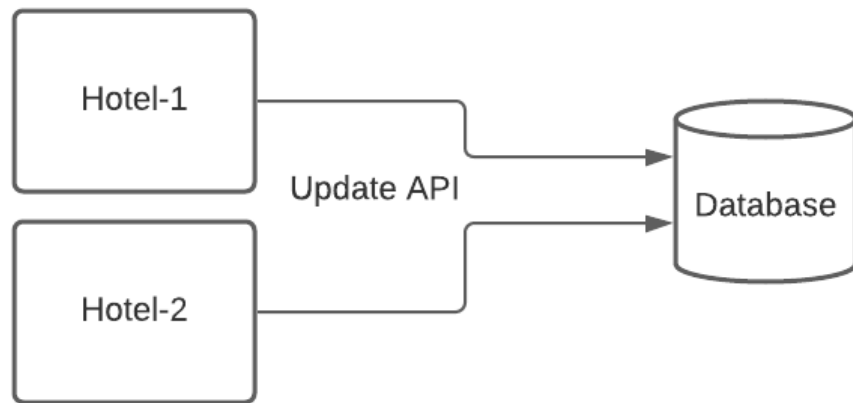
## Data flow for Fetch API-

User will use Fetch API to retrieve data from the Database.



**Data flow for Update API-**

Hotels would use Update API to update the rates/rooms available in the Database.



## 1.1 Scope

---

Scope of the following testing will be-

- Functional and regression testing of the following API
  - Fetch API
  - Update API
- Performance Testing
- End to End Testing
- UAT/Business Testing
- Execution of Automated regression suite(if available)

## 1.2 Quality Objective

---

Quality objectives of the testing are-

- Enhancing customer satisfaction
- Obtain 100% customer on-time-delivery
- Reduce production defects by 10%
- Average operation efficiency to be >90%
- Ensure the AUT meets the quality specifications defined by the client
- Bugs/issues are identified and fixed before go live

## 1.3 Roles and Responsibilities

---

Detail description of the Roles and responsibilities of different team members like

- QA
  - Cover all the testcases possible.
  - Perform all the testcases.
  - Perform regression.
  - Performance testing.
- Test Manager
  - Review of all the reports/results
- Developers
  - Cover all the cases.
- Installation Team
  - Deploy with the correct packages/versions
- Ops team
  - Monitor all the changes

## 2 Test Cases

### 2.1 Endpoints

---

- Fetch API-
  - Get – URL- {hostName}/fetch/inventory?hotel\_id=12&room\_id=123&start\_date=2021-02-19&end\_date=2021-02-21
- Update API-
  - Post- URL- {hostName}/update/inventory
- Headers-
  - None

### 2.2 Test Cases

---

- Fetch API-
  - **Success case**- All valid values
  - **Response validation**- DB value validation
  - **Hotel ID**- valid, invalid and null values
  - **Room ID**- valid, invalid and null values
  - **Start date**- valid, invalid and null values and invalid format
  - **End date**- valid, invalid and null values and invalid format
  - Not 200 response
  - With and without authorization
  - If data is not present in the DB
  - System is down

Sample request- {hostName}/fetch/inventory?hotel\_id=12&room\_id=123&start\_date=2021-02-19&end\_date=2021-02-21

Sample response-

```
{
  "hotel_id": 1,
  "room_id": 101,
  "start_date": "2021-02-19",
  "end_date": "2021-02-21",
  "data": [
    {
      "date": "2020-02-19",
      "remainingRoom": 5,
      "rate": 1 99
    },
    {
      "date": "2020-02-20",
      "remainingRoom": 3,
      "rate": 199
    },
    {
      "date": "2020-02-21",
      "remainingRoom": 0,
      "rate": 199
    }
  ]
}
```

- Update API-
  - **Success case**- All valid values- DB updated successfully
  - **Success case**- With only remaining rooms- DB updated successfully
  - **Success case**- With only rates- DB updated successfully
  - **Response validation**- DB value validation
  - **Hotel ID**- valid, invalid and null values
  - **Room ID**- valid, invalid and null values
  - **Start date**- valid, invalid and null values and invalid format
  - **End date**- valid, invalid and null values and invalid format
  - **Remaining rooms**- valid, invalid and null values
  - **Rates**- valid, invalid and null values
  - If data is not present of a specific hotelID/roomID- Insert values
  - If data is present of a specific hotelID/roomID- Update values
  - If DB is not updated- Response to be checked.
  - Not 200 response
  - With and without headers
- Performance Testing
- Regression Testing
- End to end testing with the FE-client/booking-api

Sample request -

```
{
  "hotel_id": 1,
  "room_id": 101,
  "start_date": "2021-02-22",
  "end_date": "2021-02-24",
  "data": {
    "remainingRoom": 8,
    "rate": 208
  }
}
```

Sample response -

```
{
  "success": true
}
```

## 2.3 Test Completeness

---

Test Completeness would be

- 100% test coverage
- All Manual & Automated Test cases executed
- Performance Testing executed
- Defects to be raised and closed before signoff
- All open bugs are fixed or will be fixed in next release
- Review signoff from manager/reviewer
- End to end testing completed with client.

# 3 Test Deliverables

Test Artifacts that will be delivered during different phases of the testing lifecycle.

Here are the deliverables

---

- Test Cases
  - Defect Reports
  - Performance Reports
  - End to end testing Reports
  - Customer Sign Off
- 

# 4 Resource & Environment Needs

## 4.1 Testing Tools

---

List of tools required for testing-

- Postman

- For API testing- Get and Post
- Mssql
  - DB testing- Tables/views/procedures/scripts
- Automation Tools
  - Selenium/cucumber/maven/Java
- Performance Tools
- Jenkins
  - CI/CD integration

Required to test the project

## 4.2 Test Environment

---

Following **software's** are required to for testing-

- Java- 1.8
- Maven- 3.8.1
- Jenkins-2.289.1
- Postman
- IntelliJ/Eclipse with cucumber

## 4.3 Automation

---

The following are the automation required-

- Once the functional testing is done for both the APIs using postman, add the cases to the automation suite (Regression).
- Performance tool should be automated.
- Before signoff and deployment, updated regression suite should be executed for checking regression issues on SIT/pre-prod.
- After signoff and deployment, updated regression suite should be executed again for checking regression issues on production.

# 5 Terms/Acronyms

Terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
SIT	System Integration Testing