API Testing Assessment with RestAssured and Postman

Overview

The API Testing Assessment is designed to evaluate the proficiency of candidates in testing RESTful web services using two popular tools: RestAssured (for Java-based automated testing) and Postman (for manual and automated testing).

Objectives

- Assess understanding of REST principles.
- Evaluate the ability to create and execute API test cases.
- Measure the competency in validating responses and handling various HTTP methods.
- Check the knowledge of automating API tests using RestAssured.

Practical Tasks:

- Given an API documentation, identify endpoints and methods for Create, Read and Update operations.

Postman:

Prerequisite:

- Install Postman https://www.postman.com/downloads/
- Create a new Collection

Levels of Assessment:

Level 1: Basic chaining of Requests with Hardcoded Data

Objective: Perform Create, Get, and Update in chaining ,with hardcoded data for first name and Last name in the create and update booking requests with collection runner for 3 iterations

Deliverables:

- Postman collection with Create, Get, and Update requests.
- Variable file used for execution
- Screenshots of request and response for each operation.

Level 2: Data-Driven Testing with CSV File

Objective: Perform Create, Get, and Update in chaining using data from a CSV file from for first name and last name in create and update booking requests with the Collection Runner.

Deliverables:

- Postman collection with parameterized requests.
- Variable file used for execution.
- CSV file with booking data.
- Screenshots of request and response for each operation.
- Collection Runner execution report.

Level 3: Data-Driven Testing with JSON File and Newman Reporting

Objective: Perform Create, Get, and Update operations using data for first name and last name for create and update booking from a JSON file, and generate reports using Newman.

Deliverables:

- Postman collection with parameterized requests.
- Variable file used for execution.
- JSON file with booking data.
- Newman execution report (HTML format).

Level 4: Dynamic Data Using Postman Variables

Objective: Perform Create, Get, and Update chaining operations using dynamically generated data for first name and last name using Postman variables for create and update request

Deliverables:

- Postman collection with requests utilizing Postman scripting for dynamic data.
- Variable file used for execution.
- Screenshots of request and response for each operation.
- Collection Runner execution report showing the use of dynamic data.

Assertions for Postman (Applicable for all levels of assessment)

- Assert the status code as mentioned in the API Document.
- Assert whether the response body contains the first name and last name as given in the request.
- Print the First name in the console and manually verify the whether the same matches with the first name in response.

RestAssured:

Prerequisite:

- Install Eclipse.
- Create a Maven Project.
- Add the Dependencies in pom.xml (RestAssured and TestNG).

Levels of Assessment:

Level 1: Basic chaining of Requests with Hardcoded Data

Objective: Perform Create, Get, and Update in chaining with hardcoded data for first name and Last name in the create and update booking requests

Deliverables:

- Source code of the tests.
- TestNG Report
- Screenshot of the response in console

Level 2: Data-Driven Testing with JSON File

Objective: Perform Create, Get, and Update in chaining with reading from json file for create and update booking requests

Deliverables:

- Source code of the tests.
- Input request Json file
- TestNG Report
- Screenshot of the response in console

Level 3: Generate Dynamic Data for First Name and Last Name

Objective: Perform Create, Get, and Update in chaining generating dynamic data for first name and last name using Java faker (Request body for create and update should be String and date generated from java faker should be concatenated to the request body)

Deliverables:

- Source code of the tests.
- TestNG Report
- Screenshot of the response in console

Level 4: Create Booking with Schema Validation

Objective: Validate the Response Schema for Create Booking.

Deliverables:

- Source code of the tests.
- Json Schema File
- TestNG Report
- Screenshot of the response in console

Assertions for RestAssured (Applicable for all levels of assessment)

- Assert the status code as mentioned in the API Document.
- Assert whether the response body contains the first name and last name as given in the request.
- Print the First name and last name in the console and manually verify the whether the same matches with the first name in response.
- Validate depositpaid as true in response for create booking