Cat facts Api Automation Test Strategy Document

This framework is written using Cucumber BDD, Rest Assured in Java

**Tools Required for setting up the framework:**

Java IDE (Intellij/eclipse etc)

Java

Maven

Maven dependencies for Cucumber, rest assured, open csv etc.

Test Strategy:

|  |
| --- |
| Src/test/resources/features - Cucumber feature files for testing APIs   * Easy to read and implement for any APIs |
| Src/test/java/steps –   * contains step definition for the feature file in java. * common steps for different APIs as well as API specific step definitions can be added. |
| Src/test/java/support   * Any files to support the framework is added here |
| Src/test/java/utilities   * Utility classes to manage data and tests |
| Src/test/resources/csv   * All csv files reside in this folder |

The framework was tested using a real API and feature file is added for that API as well

GET a specific detail - https://demoqa.com/BookStore/v1/Book?ISBN=9781449331818

List all details - https://demoqa.com/BookStore/v1/Books

**BDD framework** aimed at creating easy to implement feature files in **Gherkin** language for testing APIs.

Feature file can be extended to test and validate different scenarios for API status codes like 201,500,400 etc as per the requirement.

\*\***Tags**\*\* are added to facilitate execution of specific tests (regression test, smoke test etc)

\*\*Scenario outline\*\* is used to perform data driven testing

Certain steps can be \*\*reused\*\* across different feature files to test different API methods like GET, POST, DELETE, PATCH etc.

Currently cucumber reporting is used.

Logger is added.

\*\*utilities\*\* package: CSVManager class is added to manage the CSV file against which the comparison of GET and List requests are made

\*\*support\*\* package: Support classes can be added in support package to add more feature support for the framework

\*\*RequestResponseWorld\*\* class - GET and SET the api request, response, url, path objects and attributes

\*\*OpenCSV\*\* library is used to perform CSV operations

\*\*lombok\*\* library is used to generate getters and setters automatically

Currently Json response is saved in a list of hashmap and compared against the CSV Record stored in another list of hashmap

More validations and logging can be added to display the error messages if the comparison fails.

**Framework has the capability to be extended for:**

Adding reporting using extent reports or allure reports

More detailed logging for error messages

Parallel execution of feature files

Sample Jenkins file is added.

Git webhook needs to be created to trigger Jenkins pipeline after a PR is merged to a certain branch.

To run all feature files from command line or Jenkins Pipeline

\*\*mvn clean test\*\*

To run a feature file with a specific \*\*tag\*\*(example: @smokeTest), run

\*\*mvn clean test -Dcucumber.options="--tags @smokeTest"\*\*

Cucumber Test report can be found under "target/test-report/cucumber-reports.html"

Tests:

get-cat-facts.feature - Get the details stored for the fact with the specified ID - \*\*GET /facts/{id}\*\*

Json response for the specific id is retrieved.

The record for the specific id from the response is retrieved from the csv file and the json response fields is compared with csv fields

list-cat-facts.feature - List all the facts stored in the database- \*\*GET {/facts}\*\*

The fields which need to be validated is extracted from both api json response as well as CSV file and stored in a list of HashMap and compared with each other