**Fancode API Automation Framework**

This is a **BDD-based API Automation Framework** designed to validate API responses using **Cucumber** and **Rest Assured**. It ensures seamless validation by checking that all users from the city **FanCode** have completed more than half of their to-do tasks.

**Prerequisites**

* Java Development Kit (JDK)
* Maven Build tool
* Eclipse or IntelliJ any Integrated Development Environment (IDE)

**Setup**

1. Clone the repository:

[**https://github.com/AkashQA0303/DreamSports.git**](https://github.com/AkashQA0303/DreamSports.git)

1. Navigate to the Project

**cd FanCode**

1. Install the dependencies:

**mvn clean install**

1. Running the Tests

**mvn clean test or Right click on the testng.xml file and Click Run as Testng suite**

1. Test results will be displayed in the console, including Passes, Fail and Skips and any assertions.
2. Report will be generated for the result with logs

**Project Structure**

FanCode

├── src

│   ├── main

│   │   ├── java

│   │   │   ├── pojomodel

│   │   │   │   ├── Todo.java

│   │   │   │   ├── User.java

│   │   │   ├── utilities

│   │   │   │   ├── apiutils.java

│   │   │   │   ├── customlogger.java

│   ├── test

│   │   ├── java

│   │   │   ├── baseUtilities

│   │   │   │   ├── commonUtils.java

│   │   │   │   ├── ConfigReader.java

│   │   │   ├── runners

│   │   │   │   ├── TestRunner.java

│   │   │   ├── stepDefinitions

│   │   │   │   ├── tasks.java

│   │   │   │   ├── users.java

│   │   ├── resources

│   │   │   ├── features

│   │   │   │   ├── fancode.feature

│   │   │   ├── config.properties

│   │   │   ├── log4j2.xml

│   │   │   ├── allure.properties

├── target

│   ├── cucumber-report

│   │   ├── cucumber.json

│   │   ├── cucumber.html

│   ├── allure-results

│   │   ├── (Allure generated report files will be stored here)

├── pom.xml

**Framework Components**

**1. Utilities**

* **apiutils.java**: Handles reusable API operations using **Rest Assured** (e.g., GET, POST, PUT, DELETE requests).
* **customlogger.java**: Implements centralized logging using **Log4j**.

**2. Base Utilities**

* **commonUtils.java**: Contains reusable variables like baseUrl, todoList, and fanCodeUserIds that can be accessed across classes.
* **ConfigReader.java**: Reads values from the config.properties file.

**3. POJO Models**

* **Todo.java** and **User.java**: Represent API response data using **POJOs** (Plain Old Java Objects) for deserialization.

**4. Step Definitions**

* **tasks.java** and **users.java**: Implement step definitions using Cucumber for API requests, responses, and validation logic.

**5. Runner**

* **TestRunner.java**: Configured with **CucumberOptions** to define feature paths, glue code, plugins, and report generation.

**6. Resources**

* **features/fancode.feature**: Contains the BDD test scenarios.
* **config.properties**: Holds environment configurations (e.g., Base URL).
* **logs**: Stores Log4j logs.

1. **Target Folder for Reports:**

* target/cucumber-report → Contains the JSON and HTML reports generated by Cucumber.
* target/allure-results → Will contain the Allure result files (.json, .xml) after test execution.

**Reporting**

The test results are generated using ExtentReports. After test execution, the report is available at **target/**

