

# **SurePay Automation Test Report**

**Version 1.0**

**Prepared By: Unni Kannan**

## ***Introduction:***

The purpose of this document is to give high level overview about testing roadmap that will be followed for testing of SurePay Application. This document will list the detailed understanding about the Testing workflow that will be followed during Testing Life cycle.

## ***System Description:***

SurePay Test has 3 test scenarios which is scripted using Java Cucumber framework for rest api services.

## ***Scope of Testing:***

### ***In-Scope:***

#### **1) Functional Testing:**

- i. Automation testing should be done in order to reduce manual regression efforts for repetitivetests on SurePay Assignment test.

### ***Out of Scope:***

- 1) Performancetesting
- 2) Securitytesting

Tools used forautomation:

List of all the software and tools required for developing and executing automation framework.

- 1) Java (version1.8)
- 2) rest-assured
- 3) Maven
- 4) Cucumber

## ***Automation Scripts Execution Steps:***

To execute test scripts please follow below steps:

- 1) Install Java & maven and set their respective paths in systemvariables.
- 2) Clone project with below command
- 3) git clone <https://github.com/unkannan/SurepayAPIAutomation.git>
- 4) Open COMMAND PROMPT (cmd)
- 5) Navigate to the folder 'SurepayAPIAutomation
- 6) Clean and compile project using

*mvn clean compile*  
*execute the test cases using*  
*mvn test*

7) **Also the Azure pipeline is available here**

8) <https://app.circleci.com/pipelines/github/unkannan/SurepayAPIAutomation>

9) Test reports are available in below location




































*\target\reports\test-report\index.html*

### ***Automation Project structure:***

This project is implemented in Page Object Model (POM) which helps in making the code more readable, maintainable, and reusable. Mentioned below are the description about different components of framework:

- 1) ***pom.xml***: This file is used to perform build actions(clean, compile, install, test etc.) on project. All dependencies are included in pom.xmlfile.
- 2) ***endpoints***: This folder contains endpoints and request sent for endpoints.
- 3) ***model***: This folder contains the Jason model.
- 4) ***stepdefs***: This folder contains step definition files for respective feature files.
- 5) ***util***: This folder the helperclass functions.
- 6) ***resources***: This folder contains the feature files

### *Project Structure:*

- ✓  SurepayAPIAutomation
  - >  src/main/java
  - ✓  src/test/java
    - ✓  (default package)
      - >  RunCukesTest.java
    - ✓  endpoints
      - >  BaseEndpoints.java
      - ✓  PostedCommentsEndPoint.java
        - >  PostedCommentsEndPoint
    - ✓  model
      - >  PostedComments.java
    - ✓  stepdefs
      - >  CommonDefinitions.java
      - >  SearchUserPostedComments.java
      - >  World.java
    - ✓  util
      - >  HelperClass.java
  - ✓  src/test/resources
    -  SearchUserPostedComments.feature
  - >  Maven Dependencies
  - >  JRE System Library [JavaSE-1.8]
  - >  resources
  - >  src
  - ✓  target
    - >  generated-test-sources
    - >  maven-status
    - ✓  reports
      - ✓  test-report
        - >  formatter.js
        - >  index.html
        - >  jquery-1.8.2.min.js
        - >  report.js
        - >  style.css
    - >  surefire-reports
  -  pom.xml

# SurePay Automation ApplicationReport

## Test Reports

### ▼ Feature: user posted comments validation

*This feature includes tests that test the user posted comments RESTful services*

#### ▼ @Functional Scenario: Correct search results are shown for posted comments of the user id provided

**Given** the posted comments API is available

**When** I search for an user id with "1" comments details

**Then** the requests response comments has all email in correct format

#### ▼ @negative Scenario: Correct Error message is shown searching posted comments of the invalid user provided

**Given** the posted comments API is available

**When** I search for an user id with "xyz" comments details

**Then** the response status code should be "404"

```
java.lang.AssertionError:  
Expected: is <404>  
but: was <200>  
    at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20)  
    at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:8)  
    at endpoints.BaseEndpoints.verifyResponseStatusValue(BaseEndpoints.java:45)  
    at stepdefs.SearchUserPostedComments.searching_Posted_Comments_For_InvalidUser(SearchUserPostedComments.java:30)  
    at *.Then the response status code should be "404"(SearchUserPostedComments.feature:15)
```

#### ▼ @smokeTest Scenario: Correct status code is shown while searching the posted comments for valid user

**Given** the posted comments API is available

**When** I search for an user id with "1" comments details

**Then** the response status code should be "200"