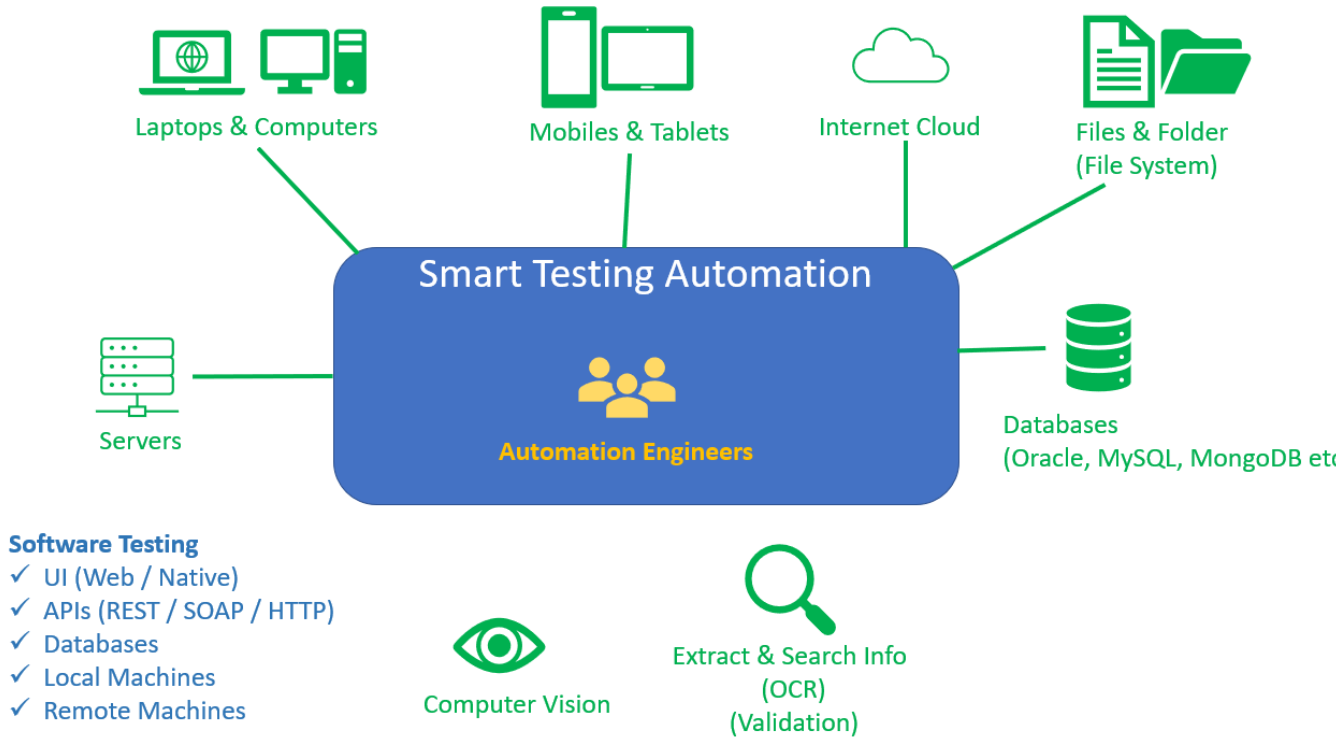

STAS CUCUMBER STEP DEFINITIONS - EN

Test Automation Developer's Guide

Abstract

This document lists the standard cucumber step definitions (English) provided by STAS tool.

Madhav Krishna

mkrishnacs20@gmail.com

Table of Contents

Table of Figures

1 Licensing and usage

This is developer's guide developed by the actual developer of Smart Test Automation Framework (STAF) / Smart TestAuto Studio tool (STAS). The purpose of this developer's guide is to help test automation engineers to understand the standard cucumber step definitions provided by STAS to automate the following types of testing: Regression Testing, Sanity Testing, Smoke Testing, End-to-End Testing, Functionality Testing, User Acceptance Testing etc.

This is a free and open-source tool and licensed under **Apache License 2.0 (https://www.apache.org/licenses/LICENSE-2.0)**. This tool and the guide only used to help develop testing automation and there is no liability on the author if there is any defect found in the system. But users can raise the defects on the Github URL (https://github.com/mkrishna4u/smart-testauto-cucumber-stepdefs-en/issues) so the community team can work on that defect and close as per their convenience. Also, if you are looking for any change or enhancement you can raise that on the same Github URL.

Also, the licensing of 3rd party tools integrated in this tool belong to the individual owner of the 3rd party tool. If there is any defect on third party tool, the defect should be raised on the respective 3rd party tool website.

2 Introduction

STAS provides the standard cucumber step definitions in English (**URL**: https://github.com/mkrishna4u/smart-testauto-cucumber-stepdefs-en) that can be used to create scenarios for testing automation. STAS definitions has connectivity with the different sub systems as mentioned in the diagram below:

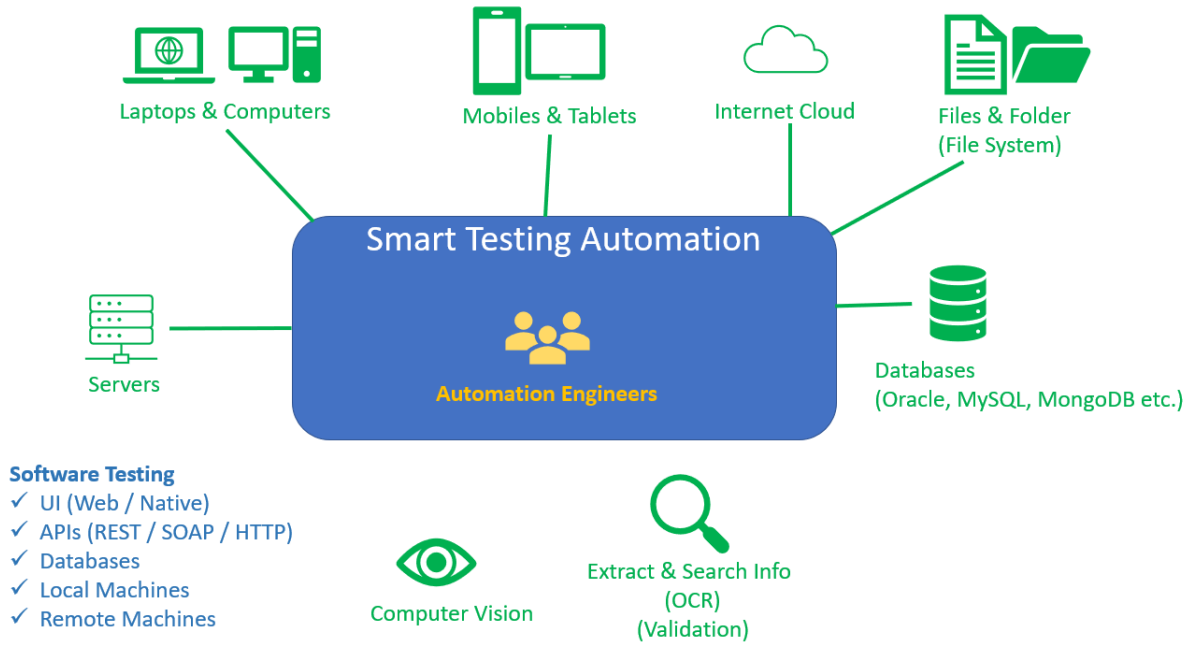

Figure 2-1 STAS Tool Connectivity

STAS tool connects with different types of applications (Web, Native, Mobile), servers (Database, file systems), remote machines (over SSH).

Using STAS, we can be able to perform the testing automation for:

A. Graphical User Interface (GUI) Testing
B. API Testing
C. Database Testing
D. Functional Testing
E. End-to-End Scenario Testing / Integration Testing (may include multiple applications and multiple sub systems)
F. Regression Testing
G. Smoke Testing
H. Cross applications testing
I. File contents verification testing i.e. PDF, MS Word, MS Excel, MS PowerPoint, Images etc.
J. File upload and download
K. OCR (Optical Character Recognition) Testing
L. Computer Vision / Image pattern matching

The following **approaches** are supported by STAS to write the test cases easily and manage them:

A. Data Driven Testing (DDT) Approach

Use different type of data file to feed the data to the system to verify the functionality. State of the art easy to use classes are provided to read the data from different types of files as given below:

1. ExcelFileReader
2. CSVFileReader
3. JsonDocumentReader
4. XmlDocumentReader
5. YamlDocumentReader
6. SmartDatabaseManager / SQLDatabaseActionHandler

B. Behavior Driven Testing (BDT) Approach

Use Cucumber Feature file to write the End-to-End scenarios and to provide the data to the system to verify the functionality.

C. Configuration Driven Testing (CDT) Approach

STAS provides standards configuration files in YAML format to configure your application related information (i.e. application config, user profile config, web driver config, database config, remote machine config etc.) that can be used to communicate with UI, REST Servers, Database Server and Remote Machines and also that can be used to validate the data on UI, REST Services, Database and Remote Machines.

STAS provides **LOW CODE / NO CODE** model, it means test engineers have to write low code or no code based on the scenario description. But they have to write scenarios in Cucumber Gherkin language as per the standardized step definitions provided by this tool.

This tool supports the **real environment** for software testing (Similar way our manual tester performs software testing like data preparation, run test cases (data-driven), data verification and generate reports etc.). Here using this tool, we can automate data preparation, test cases execution, data verification and report generation easily.

2.1 What you need to know to work on this tool?

There are many tools integrated in this tool to make the test engineers life easy. The most import things that every test engineer must know are:

1. **Cucumber Gherkin Language:** Study about it on the following URL: https://cucumber.io/docs/gherkin/reference/
2. **JSON and JSON Path:** JSON is a very powerful data format that is used while writing cucumber scenario using STAS tool. For JSON path, please refer https://github.com/json-path/JsonPath link. JSON Path is used to modify the JSON data or retrieve field data from JSON data.
3. **Basic Knowledge of Java:** Basic knowledge of Java programming is needed to create the page object classes. It only require, how to create class object using parameterized

constructor or how to call class methods. Other basic knowledge of Java programming will be useful if you are planning to write your customized cucumber step definitions.

4. **Different type of data files:** If you are writing data driven test scenarios that requires to read data from Excel, CSV, TXT, XML, JSON, YAML etc. file then you should know the format of these data files. This tool provide ready to use step definition to read the data from these types of files.
5. **XML and XPATH:** Knowledge of XML file contents are mandatory to work on the user interface and web services (API testing) that uses XML data format. Using XPATH mechanism you can access any element in XML document or modify the contents of XML document. XPATH references: https://www.w3.org/TR/1999/REC-xpath-19991116/
6. **HTML and XPATH:** For user interface testing automation, STAS tool uses Selenium / Appium internally to perform operation on page object elements like Textbox, Button etc. You can use different type of locators to locate element on the user interface like ID, AutomationID, AccessibilityID, Name, CSS Selector, LinkText, XPATH etc. By default STAS tool uses XPATH mechanism to locate element on user interface as XPATH mechanism is much solid and we can identify any element on user interface. XPATH references: https://www.w3.org/TR/1999/REC-xpath-19991116/
7. **Basic knowledge of Shell Scripting:** Since STAS uses command line to run the test scenarios. So, knowledge of how to run shell script (windows or linux) is required to run the maven commands or STAS provided scripts using command line.
8. **YAML file:** All the configuration in STAS tool is given in YAML format. It is very simple and standard format to specify the configuration. To know the YAML file format, you can refer any online document.

2.2 Supported Platform Types

The following platforms are supported to perform the software testing automation:

1. windows
2. linux
3. mac
4. android-mobile
5. ios-mobile

2.3 Supported Application Types

The following application types are supported to perform the software testing automation:

1. **native-app**: Like Desktop applications (like calculator) on any platform like windows, mac, android, iOS etc.
2. **web-app**: Like applications running on web browsers (like Github UI) on any platform like windows, mac, android, iOS etc.

2.4 Supported Web Browsers

The following web browsers are supported to perform the software testing automation:

1. chrome

2. firefox
3. edge
4. opera
5. safari
6. internetExplorer
7. remoteWebDriver
8. notApplicable: This is set for native applications.

3 How to use STAS standard cucumber step definitions?

TBD (Work in progress…)

4 Standard Cucumber Step Definitions (Provided by STAS)

TBD (Work in progress…)