Information Technology Infrastructure

Final Assignment

**SAP 38635**

Q) What is the test automation framework? What is selenium? How does it work, and why do we need it?

Ans.) **Test Automation Framework:**

A “Test Automation Framework” is scaffolding that is laid to provide an execution environment for the automation test scripts. The framework provides the user with various benefits that help them to develop, execute and report the automation test scripts efficiently. It is more like a system that has created specifically to automate our tests.

In a very simple language, we can say that a framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing. Thus, the user can follow these guidelines while automating application to take advantages of various productive results.

Types of Test Automation Framework

1. Module Based Testing Framework
2. Library Architecture Testing Framework
3. Data Driven Testing Framework
4. Keyword Driven Testing Framework
5. Hybrid Testing Framework
6. Behavior Driven Development Framework *[1]*

**Selenium:**

“Selenium automates browsers. That's it!” — SeleniumHQ

Selenium is a “test automation framework”. It is an open-source umbrella project for a range of tools and libraries aimed at supporting web browser automation. Selenium provides a playback tool for authoring [functional tests](https://en.wikipedia.org/wiki/Functional_testing) without the need to learn a test [scripting language](https://en.wikipedia.org/wiki/Scripting_language) (Selenium IDE). It also provides a test [domain-specific language](https://en.wikipedia.org/wiki/Domain-specific_language) (Selenese) to write tests in a number of popular programming languages, including [JavaScript](https://en.wikipedia.org/wiki/JavaScript) ([Node.js](https://en.wikipedia.org/wiki/Node.js)), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [Groovy](https://en.wikipedia.org/wiki/Groovy_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(software_platform)), [Perl](https://en.wikipedia.org/wiki/Perl), [PHP](https://en.wikipedia.org/wiki/PHP), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)) and [Scala](https://en.wikipedia.org/wiki/Scala_(programming_language)). The tests can then run against most modern [web browsers](https://en.wikipedia.org/wiki/Web_browser). Selenium runs on [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), and [MacOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS). It is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) released under the [Apache License 2.0](https://en.wikipedia.org/wiki/Apache_License_2.0). *[2]*

**How does selenium work?**

Selenium works with the client-server design. Client-server design is a software architecture model that consists of two parts namely a client system and a server system that communicates either through a computer network or on the same computer. For more details, let’s look at the following Selenium architecture:

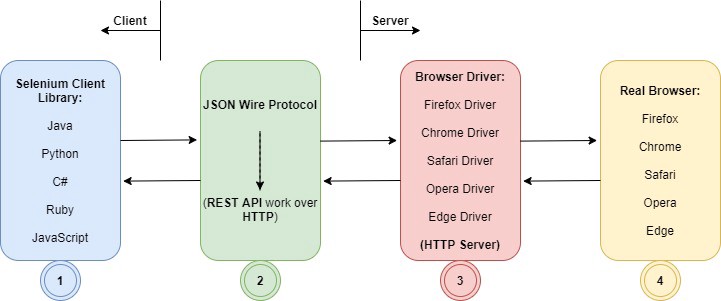


Fig.: [Selenium Architecture](https://drive.google.com/file/d/1sRSozHVOROPuLhwli22ZGrlUAcKZguZB/view?usp=sharing)

In the picture above the first point is the **Selenium Client Library** which acts as a client. **Selenium Client Library** facilitates the capability of multi-language support. We can use the language we are comfortable with to write our automation script and Selenium will do the rest.

When we write the script code in the IDE, whether in Java, C #, Python, Ruby, or Javascript, then the first thing to do is send the script to the Json Wire Protocol via API.

* **Json Wire Protocol**is at point number two in the picture and acts as a connector that facilitates communication between clients and servers so they can understand each other. **Json Wire Protocol** facilitates the ability to transfer data between clients and servers on the web. This is a REST API that provides a transport mechanism and defines RESTful web services using JSON over HTTP.

Whatever script we type in the IDE (POST/GET), Json Wire Protocol will be the connector between the client and server. So actually, Json Wire Protocol has a REST API that has json format and aims to send requests to Browser Driver via HTTP.

* At point number three is **Browser Driver**. Each browser driver has an HTTP server that accepts requests from clients sent by JWP.

Browser Driver is used to interact with the real browser and deliver automation script instructions to the real browser to be turned into an action.

* The last point is the **Real Browser**. The best part about Selenium Webdriver is that it supports all the major browsers like Google Chrome, Mozilla Firefox, Internet Explorer and Safari. Every browser has specific Webdriver for executing automation scripts. *[3]*

**Why do we need it?**

Selenium automates web browsers. It is most famous for enabling rapid, repeatable web – app testing, which allows developers to ship new releases faster and with confidence. *[4]*

Q2) The most common tools that are used for configuration management are Packer and Ansible. You need to concisely compare both of them.

Both Ansible and Packer are open source tools**.**

|  |  |
| --- | --- |
| Ansible | Packer |
| Ansible was created by Red Hat | Packer is a Hashicorp product |
| Server Configuration and Automation | Infrastructure Build Tools |
| Agentless | Cross platform builds |

**Ansible:**

 Radically simple configuration-management, application deployment, task-execution, and multi-node orchestration engine. Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates. Ansible’s goals are foremost those of simplicity and maximum ease of use.

**Packer:**

 Create identical machine images for multiple platforms from a single source configuration. Packer automates the creation of any type of machine image. It embraces modern configuration management by encouraging you to use automated scripts to install and configure the software within your Packer-made images.

Some of the features offered by Ansible are:

* Ansible's natural automation language allows sysadmins, developers, and IT managers to complete automation projects in hours, not weeks.
* Ansible uses SSH by default instead of requiring agents everywhere. Avoid extra open ports, improve security, eliminate "managing the management", and reclaim CPU cycles.
* Ansible automates app deployment, configuration management, workflow orchestration, and even cloud provisioning all from one system.

On the other hand, Packer provides the following key features:

* Super fast infrastructure deployment. Packer images allow you to launch completely provisioned and configured machines in seconds, rather than several minutes or hours.
* Multi-provider portability. Because Packer creates identical images for multiple platforms, you can run production in AWS, staging/QA in a private cloud like OpenStack, and development in desktop virtualization solutions such as VMware or VirtualBox.
* Improved stability. Packer installs and configures all the software for a machine at the time the image is built. If there are bugs in these scripts, they'll be caught early, rather than several minutes after a machine is launched.

**DigitalOcean**, **9GAG**, and **Rainist** are some of the popular companies that use Ansible,

whereas Packer is used by **Instacart**, **Oscar Health**, and **Razorpay.** *[5]*

*References:*

1. <https://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/#Test_Automation_Framework>
2. <https://en.wikipedia.org/wiki/Selenium_(software)>
3. <https://medium.com/easyread/how-selenium-works-1414c3ac5d3f>

1. <https://www.browserstack.com/selenium#:~:text=Selenium%20automates%20web%20browsers%20and,desktop%20browsers%20%26%20real%20mobile%20devices>.

1. <https://madeintandem.com/blog/packer-ansible-terraform-devops-immutable-servers/#:~:text=Packer%20supports%20multiple%20%E2%80%9Cprovisioners%2C%E2%80%9D,configuration%20of%20the%20virtual%20machine>.