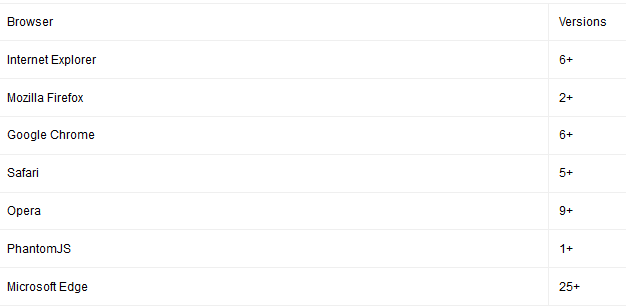
Aleksandar Zoric

How does it work?

# Introduction

Before Sahi is explained on how it works, some main features that this tool for automation contains will be listed first. There is two versions of Sahi, the open source free product and Sahi pro which is the commercial version. Sahi is best suited for multi-browser testing that have a complex 2.0 applications and also contain AJAX and dynamic content. Sahi can run on any browser that supports JavaScript. Some of the most popular browsers used that Sahi supports are shown in the below 1.1 Fig;

***Fig 1.1*** 

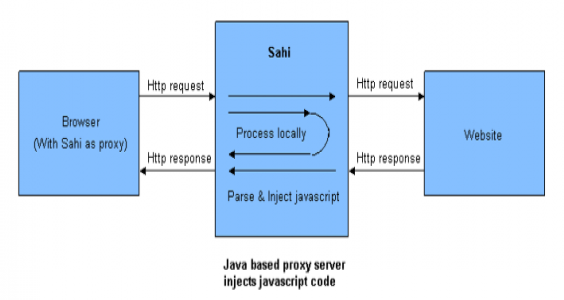
(Tyto Software Ltd, 2016)

AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating interactive, dynamic web applications using XML, HTML, CSS and JavaScript. XHTML is used for content, CSS for presentation, JavaScript for dynamic content display, JavaScript is also used to make a request to the serve, interpret the results and update the current screen the user is viewing. XML then is used for receiving server data (TutorialsPoint, 2015)

# How does Sahi work?

Once the user has downloaded and installed their chosen version of Sahi, they can then start using Sahi to test their desired web application. Sahi’s core features include the Sahi proxy server and the JavaScript Engine. Below is a diagram that shows how Sahi works using these core features;

***Fig 1.2***

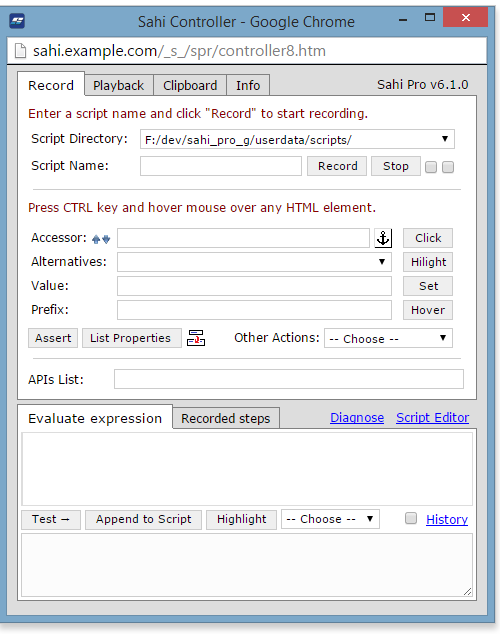
****** (ThoughtWorks, 2011)

HTML responses that pass through the proxy are amended so that JavaScript is injected at the start and the end of the response. This process then allows the chosen browser to record and playback scripts and talk back to the proxy when needed. Basically, the process above keeps track of all the communication occurring which can be later referred to if needed by the browser.

The scripts can be written in any text editor that supports recording by the controller, controller being a feature Sahi contains. If the recording is setup, what it then does is record any action that happens on the page. And it also adds some other functionality to help the user. From here the user can also add assertions, which are added to the script. The script once saved has a .SAH extension (ThoughtWorks, 2011)

Sahi controller is an IDE that is used to configure Sahi/browser options. A sahi controller is displayed as such;

***Fig 1.3***



(Tyto Software Ltd, 2016)

So once the controller is configured, the user starts to record and perform some actions on the webpage. After the recording has been stopped, all the steps executed by the user are recorded and saved. The user can then choose the playback the recorded session, which then executes the script and displays the executed steps and logs them into a database. These logs can then be viewed at any point (Tyto Software Ltd, 2016).

So scripts are all based on JavaScript but each script is parsed by Sahi i.e. some modifications added and script saved as .SAH file. Using these scripts we can write anything that JavaScript is capable of as Sahi inherits every possibility from JavaScript e.g. functions. Once a script is created by the user recording some actions on the chosen webpage, once we play it back using the Sahi controller, each step will be identified as either success or failure. Assertions are used for verification. This is where the user specifies which path or what should be the correct outcome.

Step by step execution;

1. Record actions on the webpage
2. Modify script adding assertions, functions etc.
3. Stop recording. Script saved.
4. Run playback of the script
5. Sahi generates a log which displays some useful information along with either pass or fail of each action.

\*Add to PowerPoint slide – Key points to talk about

* Sahi is best suited for multi-browser testing that support JavaScript - Add fig 1.1
* Sahi’s core features include the Sahi proxy server and the JavaScript Engine – Add fig 1.2
* The scripts can be written in any text editor
* Sahi controller is an IDE that is used to configure Sahi/browser options. - Add fig 1.3
* Scripts are all based on JavaScript but each script is parsed by Sahi i.e. some modifications added and script saved as .SAH file.
* Step by step execution;

1. Record actions on the webpage
2. Modify script adding assertions, functions etc.
3. Stop recording. Script saved.
4. Run playback of the script
5. Sahi generates a log which displays some useful information along with either pass or fail of each action.