

8.5 GB
Flash Layer Free

Xiaomi Mi 3



Blazing fast, nifty features, low price

Nikon Coolpix P530



A solid Prosumer Camera with 42x Zoom

HTC One E8



The affordable cousin of the flagship HTC One M8

₹150

www.pcquest.com

PCQUEST

UNDERSTAND • CHOOSE • IMPLEMENT IT

OCTOBER 2014

Subscriber copy. Not for sale

Network Vulnerability Testing with OpenVAS

Virtualize a Bare Metal Machine with VMware ESXi 5



Tablets

Laptops

Smartphones

Cameras

Printers and MFDs

External Storage

Desktop Anti-virus SW

Bluetooth Wireless Speakers

Headphones

Monitors

Peripherals and Accessories

Shoppers' Special:

150+ Products for the Festive Season



Reviews, comparisons, and essential buying advice to help you make wise purchase decisions

Hot Trends:

- Retail Analytics for Better Customer Relations
- Handling Privacy and Security Concerns with Big Data
- Mobile Analytics Unleashed
- The Tech Behind UHD Displays

Special Feature:

Harnessing the Power of Internet of Things for your Industry

Developer:

Automate Web App Testing with Sahi



Automating Web App Testing Using Sahi

Sahi is a free and Open Source tool that automates the process of web application testing

- Raj Kumar Maurya

It is available both as a free product and as Sahi Pro, the commercial version. This tool is used by testing teams in product companies and captive IT units. Sahi is used for cross-browser/multi-browser testing of complex web 2.0 applications with lots of AJAX and dynamic content. It runs on any modern browser which supports JavaScript.

Prerequisites

You need Java 1.5 or above for running Sahi.

Installing Sahi

Step 1: Download the software from their site and double click on it to install it. If Java is configured properly, it will launch the installer.

Step 2: If the installer does not launch, or if the jar file is opened as if it is a zip file, cancel the unzip dialog. Open a command prompt, cd to where the file is downloaded, and run by following command

```
java -jar install_sahi_pro_xxx.jar
```

Step 3: Install Sahi in a folder path without spaces. (If there are spaces, Chrome will have trouble launching correctly.)

Step 4: Start Sahi Dashboard by double clicking on the desktop shortcut in Windows and in Linux Go to

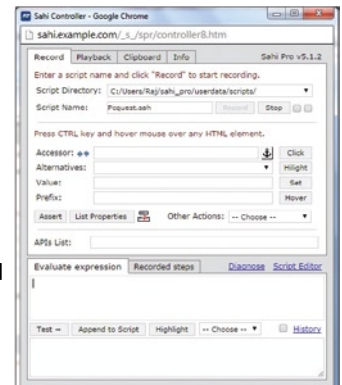
```
<sahi_root>/userdata/bin and run ./start_dashboard.sh
```

The Sahi Dashboard automatically starts the Sahi proxy, and allows launching of configured browsers. Sahi looks for browsers in their default install location, and if found, displays it on the Dashboard.



Step 5: Bring up the browser of your choice by clicking on any of the browser icons. You should see a browser window like the one shown here. We have selected Google chrome.

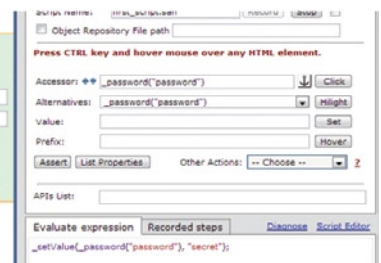
Step 6: Press *Alt* and double click on the browser window. Sahi's Controller window will pop up. If that does not work, press *Ctrl* and *Alt* keys together and then double click. Make sure popup blockers are turned off. On the Controller, go the Record tab which records your session.



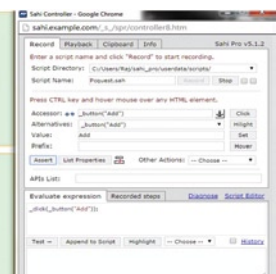
1. Give [Pquest.sah](#) or the name or your choice for the script, and click Record.
2. On the browser, you get an option like the one shown here. For testing purpose, we click on the Sample Application link on Sahi's start page.
3. This will bring up Sahi's sample application.

Step 7: Enter username 'test' and password 'secret' and click Login

The last recorded step will be visible in the Evaluate Expression box.



Step 8: On the next page, which shows a shopping cart, add quantities 2,3,1 and click Add. This will display the added items at the bottom and its total.



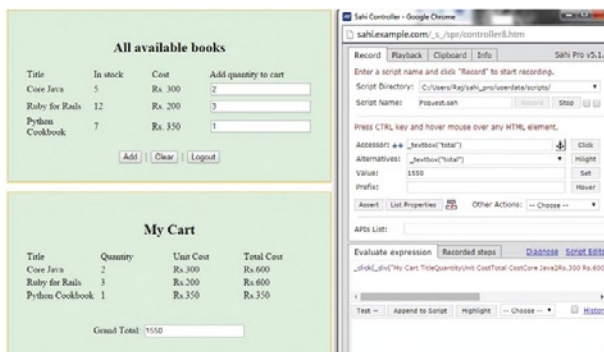
Add assertions

Step 9: A script normally consists of actions performed on the web page and then verification of functionality. For verifications, we use assertions. Sahi allows adding assertions while recording. Follow these steps to add an asser-

[Logs](#) | [Online Documentation](#)
[Test Pages](#) | [Sample Application](#)

tion on the total field:

1. Move the mouse over any html element while pressing *Ctrl* key.
2. The Accessor field will get populated in the Controller.
3. In this case, we hover over the text field near Grand Total.
4. Click the Assert button to generate assertions for the element.
5. They will appear in the Evaluate Expression box.
6. Click Test --> to check that the assertions are true.

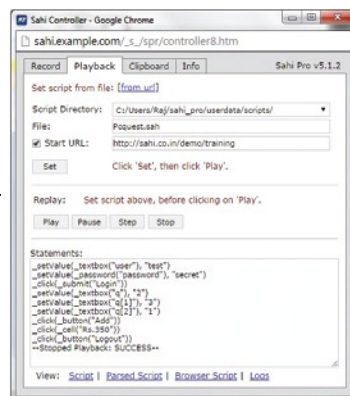


7. Once satisfied, click on *Append to Script*. This will add the assertions to the Script.
8. Click *Logout* to logout of the application.
9. Click *Stop* on the Controller to finish recording.
Note that the Controller can be closed and reopened at any time, without disrupting the recording.

Step 10: Playing back

Enter the script name. In our case, it is Pcquest.sah in the File field. It can also be filled with the help of the auto completion feature.

1. Enter the Start URL source e.g. <http://www.xyz.com>. In our case it is <http://sahi.co.in/demo/training/>.
2. Click *Set*.
3. Wait for the page to reload.

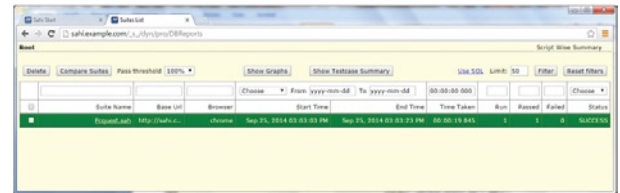


4. Click *Play*.

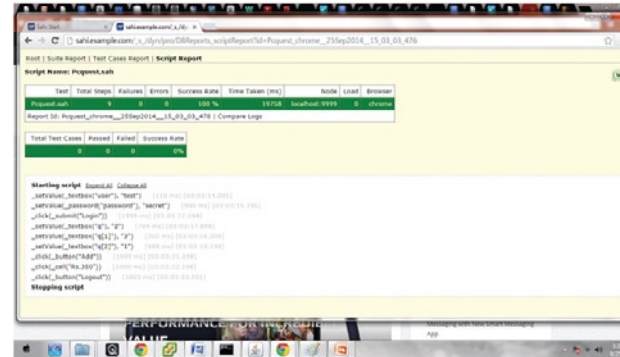
The system will start executing the steps, and the Controller will be updated accordingly. Once finished, SUCCESS or FAILURE will be displayed at the end of the steps.

Step 11: View Logs

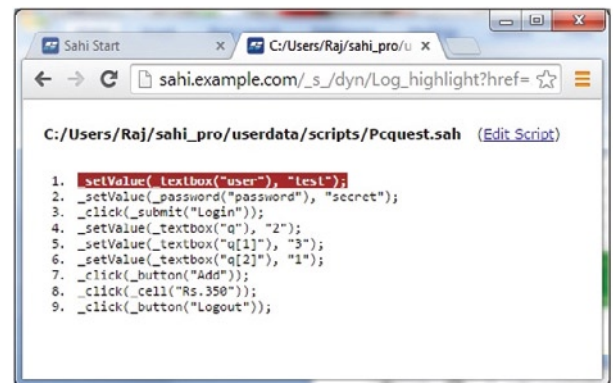
In the step 11, On the Controller, go to *Playback* tab and click on *View Logs* link at the bottom right. It will open a window with the results in html.



To view detailed logs of any script, select the one which you want to see as in the image here.



If you want to see the exact line in script, click on a line in the logs which will further open up the exact line of the script. You can also view the logs at <http://localhost:9999/logs>.



Sahi is the tool that can run scripts in parallel on multiple instances of Internet Explorer. Sahi works even when elements do not have IDs. Sahi uses relational APIs like `_in`, `_near`, `_under`, `_leftOf`, `_rightOf` etc. to uniquely identify elements. Sahi automatically waits for page loads and ajax activity and you can use it for testing more things as well. □