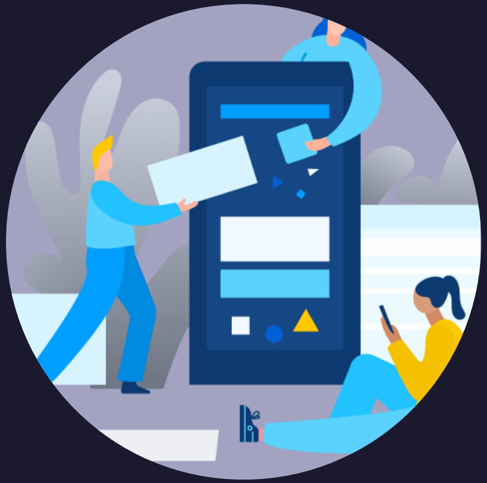
Sahi Software

Group 11

Agenda



Introduction

- Sahi is a software testing tool that automates web application testing. It was developed by Tyto Software in 2005 as an open-source tool, and later commercialized with the release of Sahi Pro in 2010.

- Sahi Pro provides a range of features to support web application testing, including record and playback, script editing, test case management, and reporting.

- Sahi can be used to test Web Browsers, Desktop Applications, and Mobile Apps

Features

Sahi is a software testing tool that provides a range of features to support testing.

Sahi has Simple & Powerful APIs to easily identify elements on DOM, perform mouse, keyboard and touch actions seamlessly.

Sahi embeds automatic waits and eliminates wait for statements and can navigate inconsistent page loads.

Sahi Object Spy and Recorder to identify elements across devices and software. It can also work on applications that generate dynamic IDs for elements.

Features

Sahi implements Business friendly frameworks and uses inbuilt Business Driven Test Automation to let business analysts and non-technical professionals contribute towards test cases.

Sahi employs automatic logging while automating applications that contains the complete information about the run. This helps testers identify the exact line where script is failing. All logs are stored in database.

Sahi can run thousands of scripts that are packed into a suite on a single machine in parallel mode or can be distributed across machines.

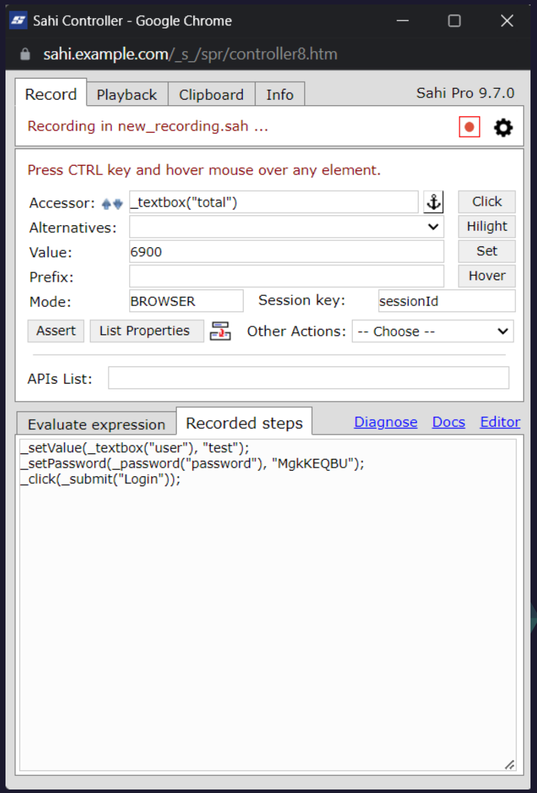
Roles and Input Combinations

- Sahi is suitable for use by a range of stakeholders involved in software testing, including developers, testers, and QA engineers
- Sahi uses control + hover combination to identify elements in DOM. Uses Sahi Script to store the user events and is an extension of JavaScript, capable of interacting with browser efficiently and perform reads on file system, access database and can call Java etc.,

Roles and Input Combinations (Contd.)

- Sahi Pro offers a variety of ways to identify web elements on a page, including through traditional methods such as ID, name, and CSS selectors, as well as advanced techniques like relation APIs (such as near, in, under, leftOf, and rightOf) that allow testers to identify elements with respect to each other.

- Sahi is Browser and Operating System independent in most cases. Sahi uses proxy to inject Javascript into pages. In 95% of the test cases, this will work. In some special cases, where Javascript events are ineffective, Sahi falls back to native events. These require focus on test window and prevents parallel testing.



Recording in new_recording.sah ...

Press CTRL key and hover mouse over any element.

Accessor:	_textbox("total")
Alternatives:	
Value:	6900
Prefix:	
Mode:	BROWSER
Session key:	sessionId
Other Actions:	-- Choose --
APIs List:	

```
_setValue(_textbox("user"), "test");
_setPassword(_password("password"), "MgkKEQBU");
_click(_submit("Login"));
```

Economics

- Sahi started as Open-Source project in 2005 with focus on automation of emerging web 2.0 technologies
- Sahi evolved into Sahi Pro that handles automation on modern web browsers, mobiles, and desktop applications

Sahi Pro has 4 pricing plans available for customers

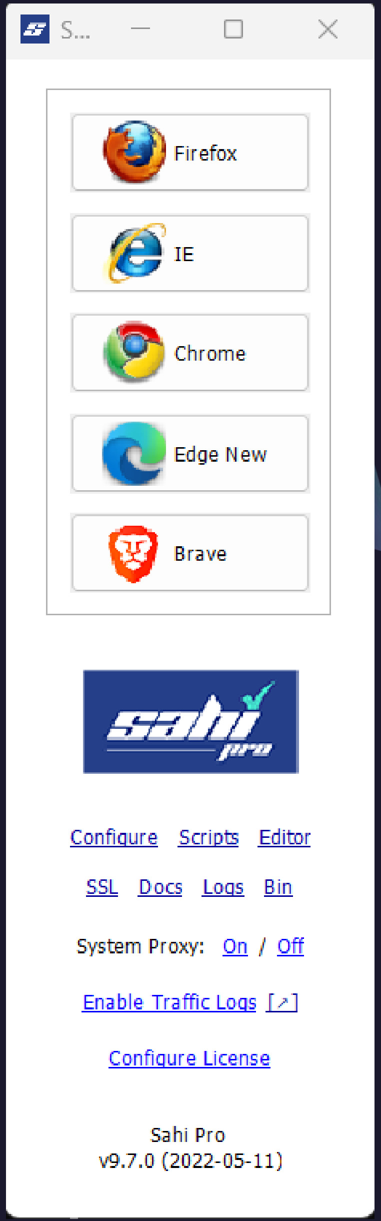
- **Sahi Pro for Web**
  - Works across browsers and operating systems
- **Sahi Pro for Desktop**
  - Works for windows desktop applications(WPF, .NET, etc.,)
  - Java based applications
- **Sahi Pro for Mobile**
  - Mobile native & hybrid applications in iOS and Android
- **Sahi Pro for SAP**
  - SAP GUI for windows

All plans support Database, File system, REST/Web services
Each plan has user / concurrent licenses available.

Sahi Pro offers 30 days free trial.

Interface

- Sahi uses Java to run and execute application.
- After starting it lists browsers present on the system.
- User can select the browser to open web interface of Sahi.
- User can press the ALT key and Double Click on page to bring up the Sahi Controller
- Sahi Scripts can be recorded and played back from the Controller.
- The controller allows you to identify elements, create assertions, execute code snippets, record sequences, playback scripts and gives access to various functionalities of cipher.



Sahi Pro
v9.7.0 (2022-05-11)

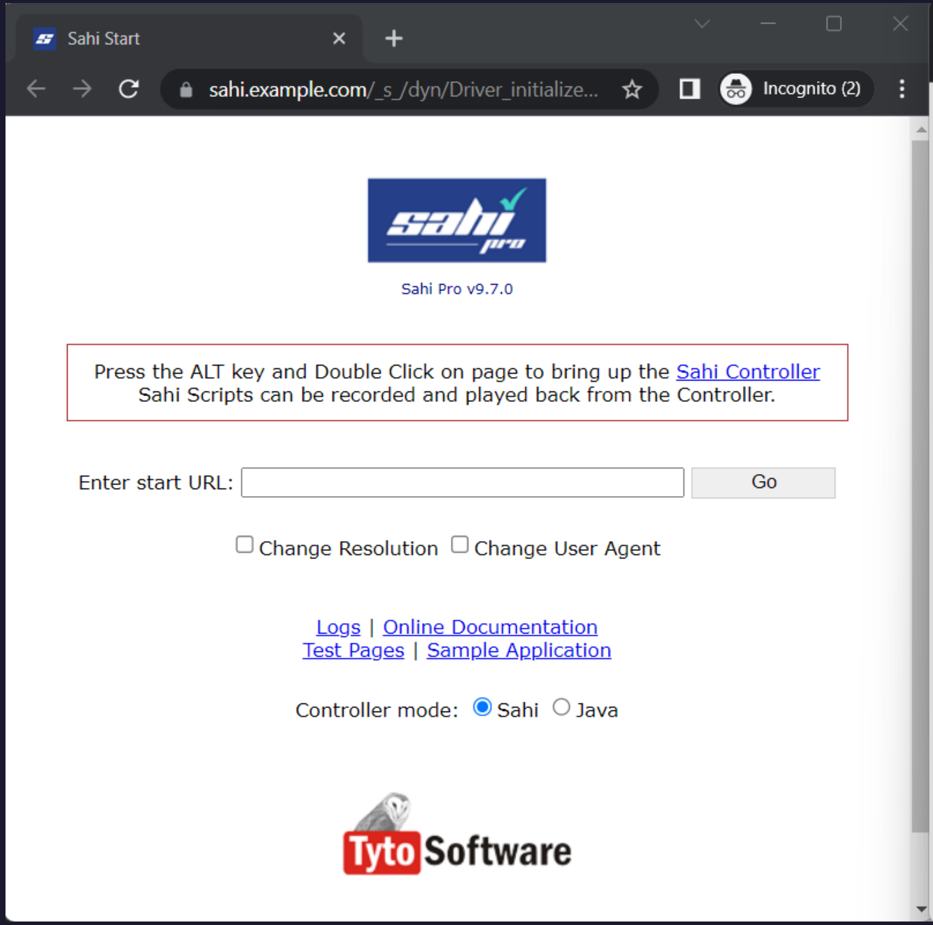
Sahi Controller

Sahi Controller has Record, Playback, Clipboard and Info menus to help in automation process.

Users can provide a script name with extension .sah and click on red dot button to start recording user actions on an application under test.

All actions are shown in Evaluate Expression and Recorded steps tabs to see the steps being recorded.

Clicking the red dot button will stop the recording and user can see recorded steps in Recorded steps tab of controller.


Sahi Start

sahi.example.com/_s_/dyn/Driver_initialize...

Incognito (2)

Sahi Pro v9.7.0

Press the ALT key and Double Click on page to bring up the Sahi Controller
Sahi Scripts can be recorded and played back from the Controller.

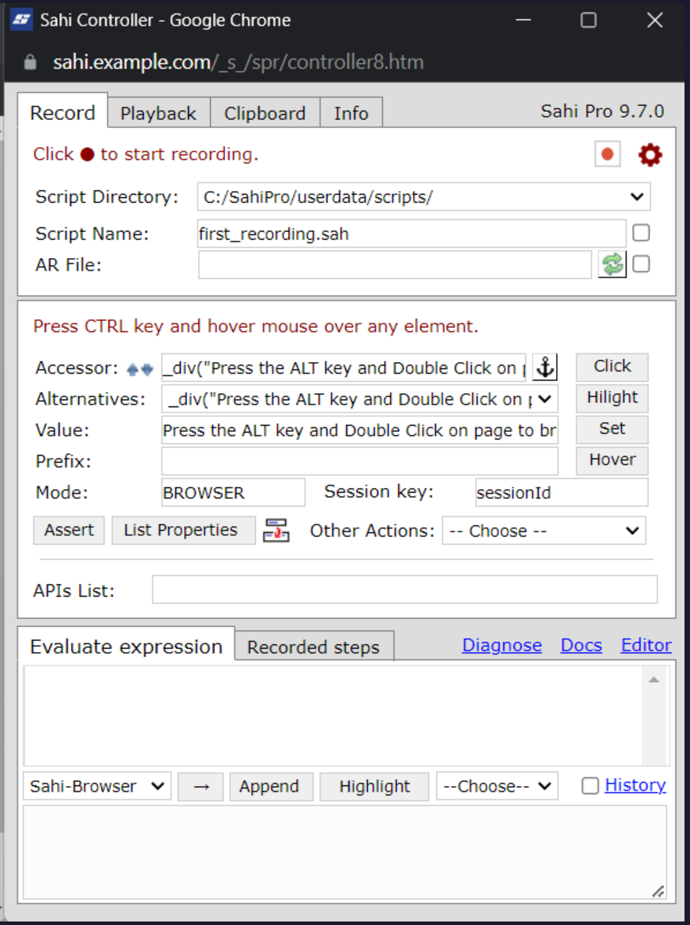
Enter start URL: Go

☐ Change Resolution ☐ Change User Agent

Logs | Online Documentation
Test Pages | Sample Application

Controller mode: [✓ Sahi] Java

Tyto Software


Sahi Controller - Google Chrome

sahi.example.com/_s_/spr/controller8.htm

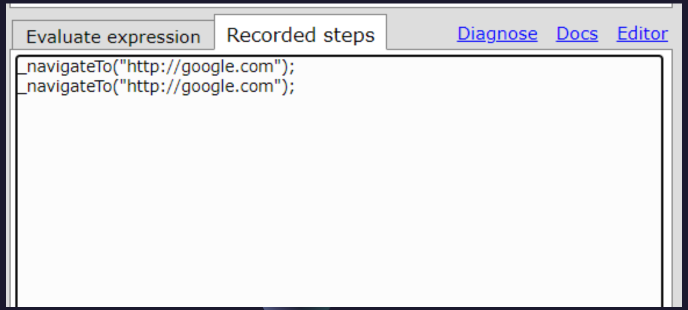
Record | Playback | Clipboard | Info — Sahi Pro 9.7.0

Click ● to start recording.

Script Directory:	C:/SahiPro/userdata/scripts/
Script Name:	first_recording.sah
AR File:	

Press CTRL key and hover mouse over any element.

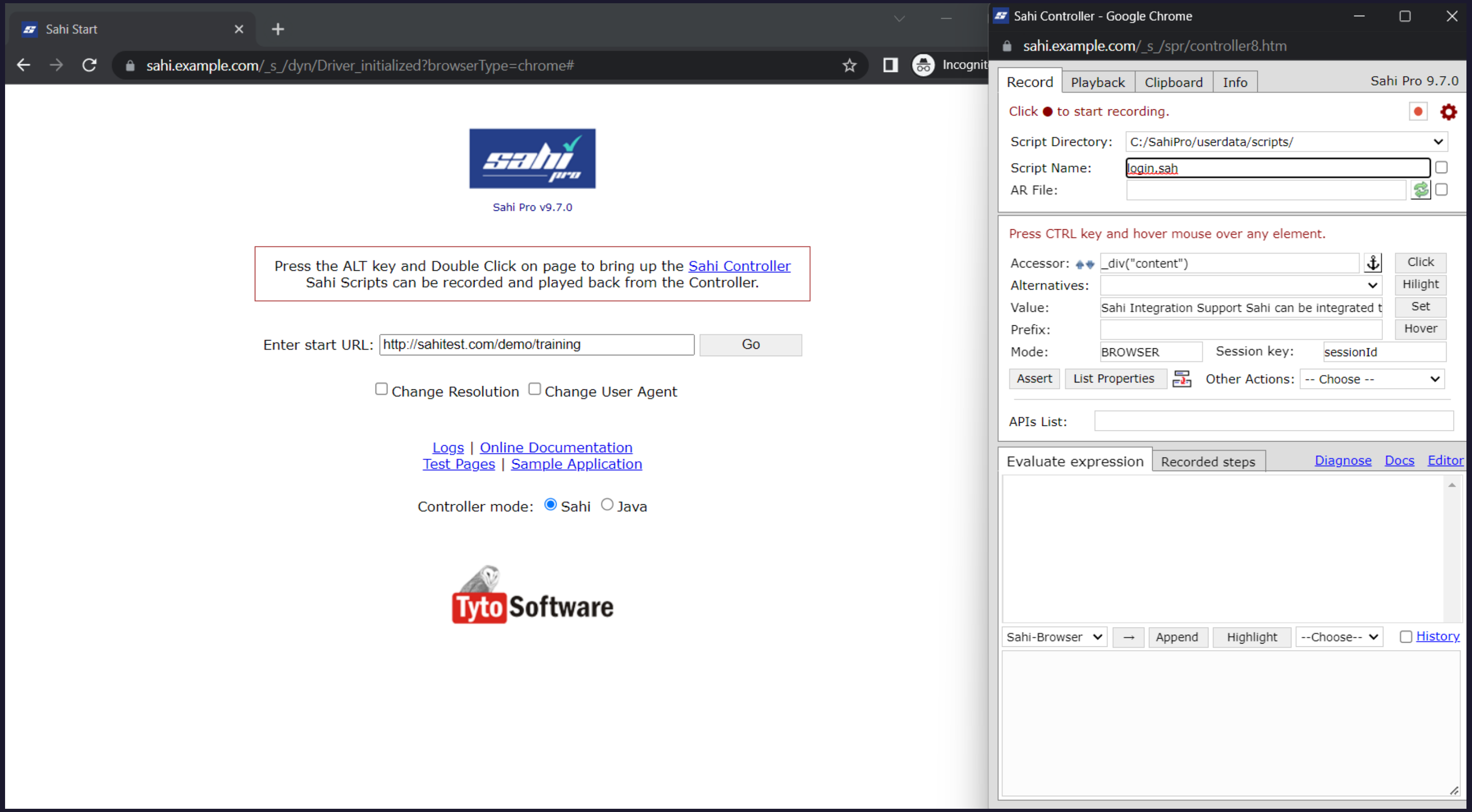
Accessor:	_div("Press the ALT key and Double Click on
Alternatives:	_div("Press the ALT key and Double Click on
Value:	Press the ALT key and Double Click on page to br
Prefix:	
Mode:	BROWSER
Session key:	sessionId
Other Actions:	-- Choose --
APIs List:	



```
_navigateTo("http://google.com");
_navigateTo("http://google.com");
```

Record and Playback

- Let us start by recording and playing back a simple script.
- Sahi Pro is not just a record and playback tool, recording is a steppingstone to creating automation scripts.
- User can start application by clicking the application shortcut on desktop.
- Let's record our first script.
- Enter a script name to save the recording. And click record button.



Sahi Start

sahi.example.com/_s_/dyn/Driver_initialized?browserType=chrome#

Sahi Pro v9.7.0

Press the ALT key and Double Click on page to bring up the Sahi Controller
Sahi Scripts can be recorded and played back from the Controller.

Enter start URL: http://sahitest.com/demo/training Go

Change Resolution Change User Agent

Logs | Online Documentation
Test Pages | Sample Application

Controller mode: [✓ Sahi] Java

Sahi Controller - Google Chrome

sahi.example.com/_s_/spr/controller8.htm

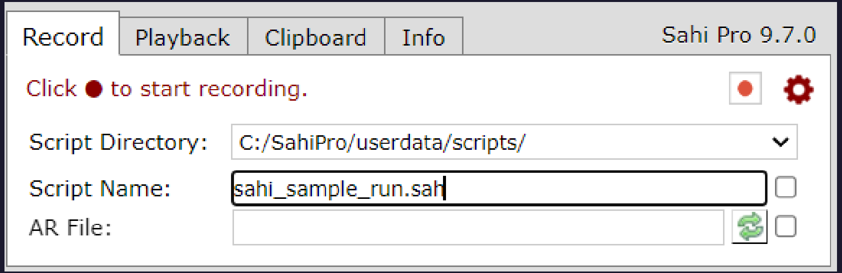
Record | Playback | Clipboard | Info — Sahi Pro 9.7.0

Click ● to start recording.

Script Directory: C:/SahiPro/userdata/scripts/
Script Name: login.sah
AR File:

Press CTRL key and hover mouse over any element.

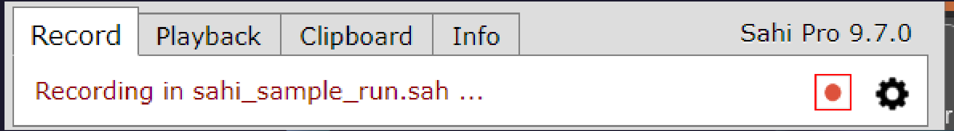
Accessor: _div("content")
Alternatives:
Value: Sahi Integration Support Sahi can be integrated t
Prefix:
Mode: BROWSER Session key: sessionId
Other Actions: -- Choose --
APIs List:



Record | Playback | Clipboard | Info — Sahi Pro 9.7.0

Click ● to start recording.

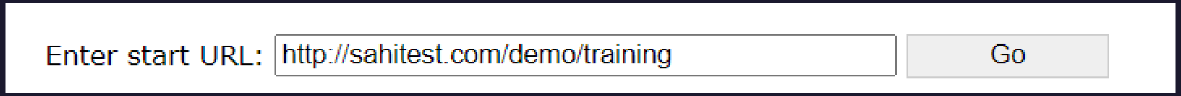
Script Directory: C:/SahiPro/userdata/scripts/
Script Name: sahi_sample_run.sah
AR File:

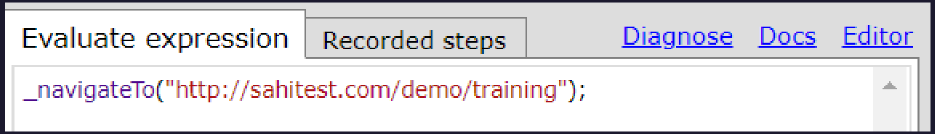


Record | Playback | Clipboard | Info — Sahi Pro 9.7.0

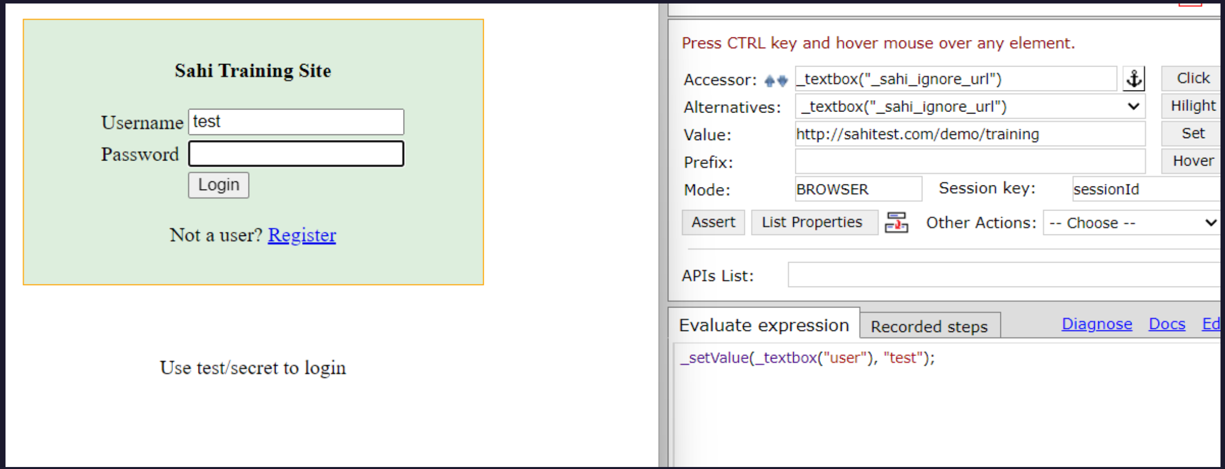
Recording in sahi_sample_run.sah ...

- Ener a start URL: http://sahitest.com/demo/training to open Sahi provided demo application to test.
- Click on Go button to start recording steps in application.
- User can see the navigateTo expression being recorded onto script.
- When user types in username, the event is recorded.
- All the steps are autosaved to the file.
- User now clicks on login button, to record the login action.


Enter start URL: http://sahitest.com/demo/training



```
_navigateTo("http://sahitest.com/demo/training");
```


Sahi Training Site

Username: test
Password:

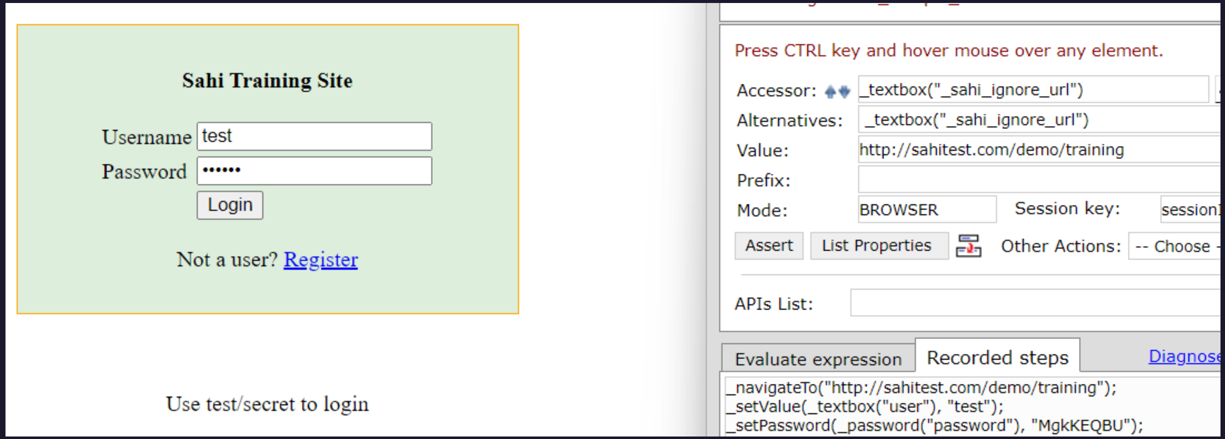
Not a user? Register

Use test/secret to login

Press CTRL key and hover mouse over any element.

Accessor: _textbox("_sahi_ignore_url")
Alternatives: _textbox("_sahi_ignore_url")
Value: http://sahitest.com/demo/training
Prefix:
Mode: BROWSER
Session key: sessionId
Other Actions: -- Choose --
APIs List:

```
_setValue(_textbox("user"), "test");
```


Sahi Training Site

Username: test
Password: ••••••

Not a user? Register

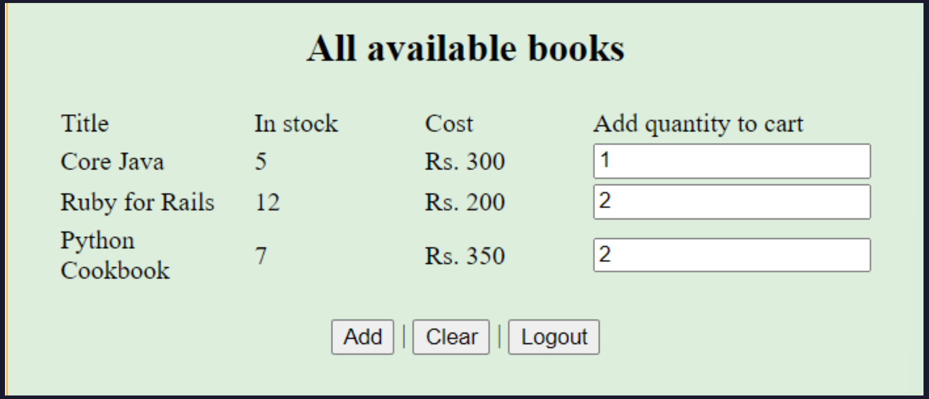
Use test/secret to login

Press CTRL key and hover mouse over any element.

Accessor: _textbox("_sahi_ignore_url")
Alternatives: _textbox("_sahi_ignore_url")
Value: http://sahitest.com/demo/training
Prefix:
Mode: BROWSER
Session key: session
Other Actions: -- Choose
APIs List:

```
_navigateTo("http://sahitest.com/demo/training");
_setValue(_textbox("user"), "test");
_setPassword(_password("password"), "MgkKEQBU");
```

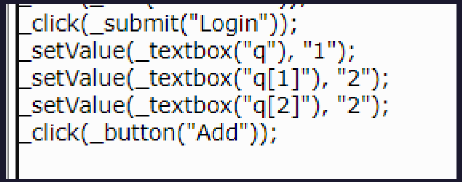
Now we add books quantity to cart



All available books

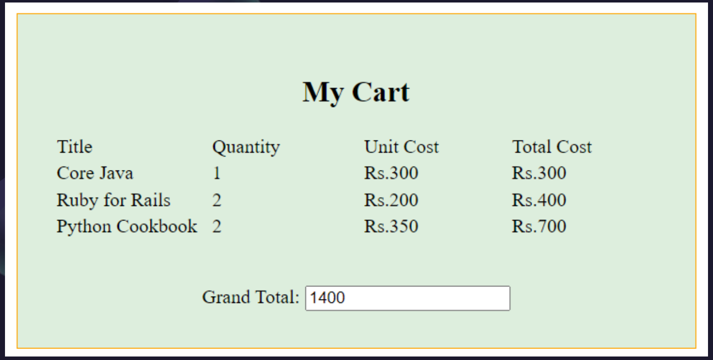
Title	In stock	Cost	Add quantity to cart
Core Java	5	Rs. 300	1
Ruby for Rails	12	Rs. 200	2
Python Cookbook	7	Rs. 350	2

All the actions are recorded in Recorded steps view



```
_click(_submit("Login"));
_setValue(_textbox("q"), "1");
_setValue(_textbox("q[1]"), "2");
_setValue(_textbox("q[2]"), "2");
_click(_button("Add"));
```

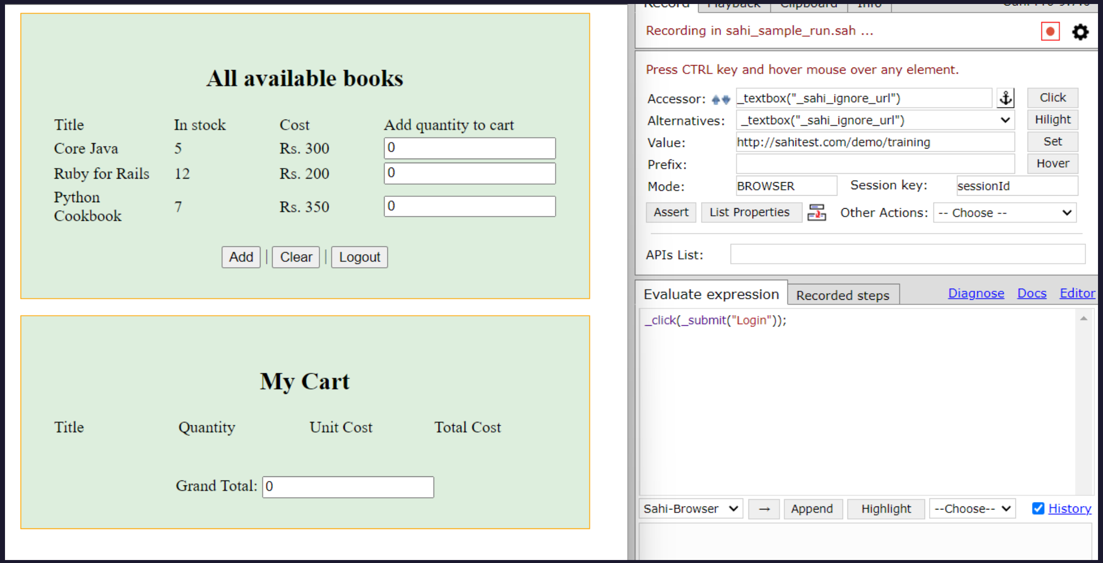
After Add button is clicked, books are added to the Cart



My Cart

Title	Quantity	Unit Cost	Total Cost
Core Java	1	Rs.300	Rs.300
Ruby for Rails	2	Rs.200	Rs.400
Python Cookbook	2	Rs.350	Rs.700

Grand Total: 1400



All available books

Title	In stock	Cost	Add quantity to cart
Core Java	5	Rs. 300	0
Ruby for Rails	12	Rs. 200	0
Python Cookbook	7	Rs. 350	0

My Cart

Title	Quantity	Unit Cost	Total Cost

Grand Total: 0

Recording in sahi_sample_run.sah ...

Press CTRL key and hover mouse over any element.

Accessor: _textbox("_sahi_ignore_url")
Alternatives: _textbox("_sahi_ignore_url")
Value: http://sahitest.com/demo/training
Prefix:
Mode: BROWSER Session key: sessionId
Other Actions: -- Choose --
APIs List:

```
_click(_submit("Login"));
```

In this test case we verify that the Grand Total is correct, for the books quantity we added to cart.

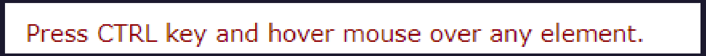
We identify the total element and create an assertion for it. To do that we need to follow the step shown in controller.

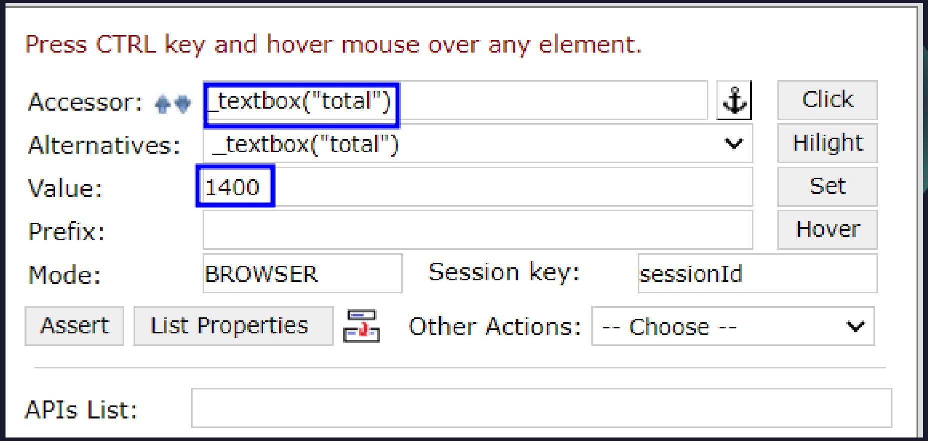
Press Control key and hover mouse over total element.

User can now click on Assert button to get different assert suggestions that can be done on the selected element.

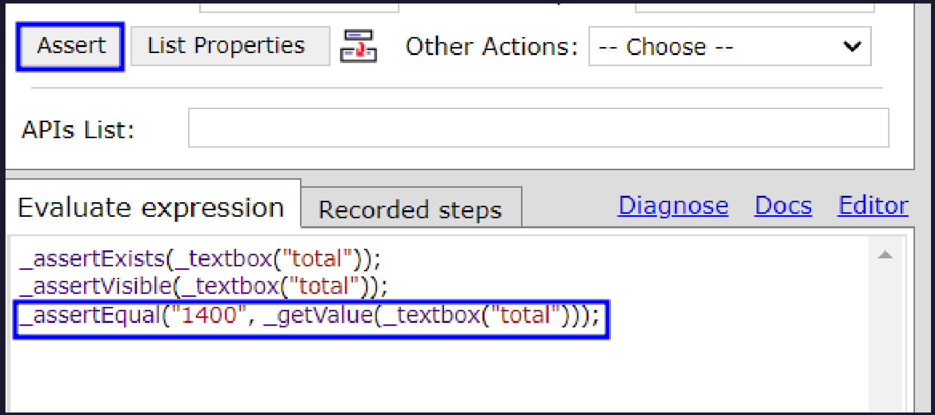
Generated assert statements check if the element is existing in DOM, visible to user and verifies if it is equal to 1400 value.

User can verify the assertions by clicking the arrow button below the Evaluate expression tab.

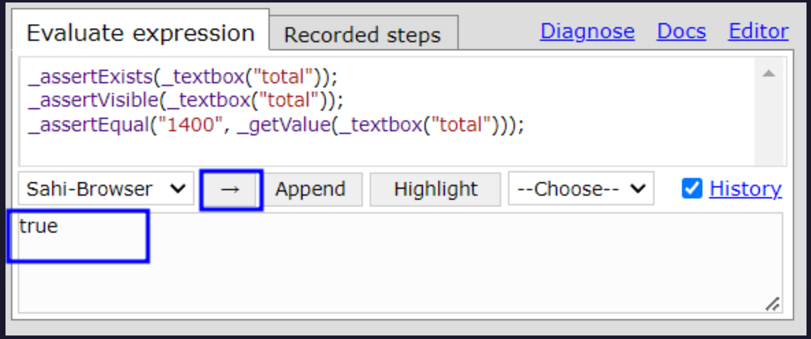

Press CTRL key and hover mouse over any element.


Press CTRL key and hover mouse over any element.

Accessor: _textbox("total")
Alternatives: _textbox("total")
Value: 1400
Prefix:
Mode: BROWSER
Session key: sessionId
Other Actions: -- Choose --
APIs List:


Other Actions: -- Choose --
APIs List:

```
_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1400", _getValue(_textbox("total")));
```

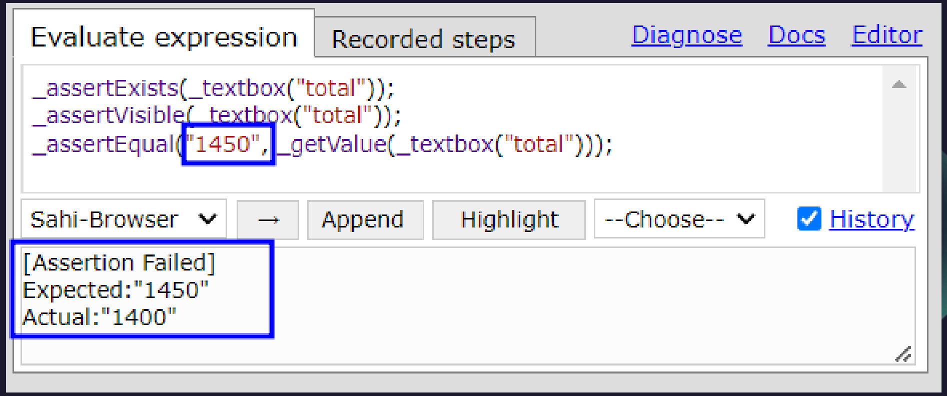


```
_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1400", _getValue(_textbox("total")));
```

Sahi-Browser → Append Highlight --Choose-- ✓ History

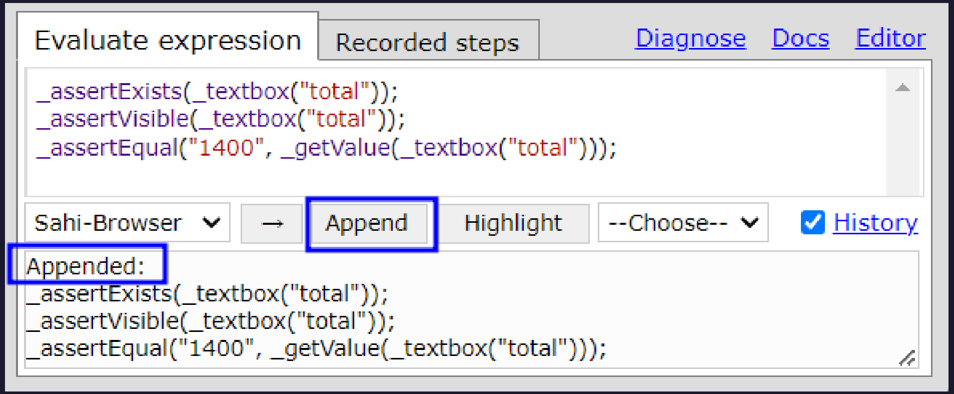
```
true
```

- Changing 1400 value to 1450 and then evaluating the asserts will result in test failure.
- When we are good with the assert statements, clicking on Append button appends the generated asserts to the script.
- Actions that we perform on the browser are automatically recorded. Actions that we perform on the Sahi Controller, need to be appended to the script manually.
- Let's logout, examine the recorded script and run the automation from start.
- Clicking red button will stop the recording and save all recorded steps to the .sah file provided at the start of recording.



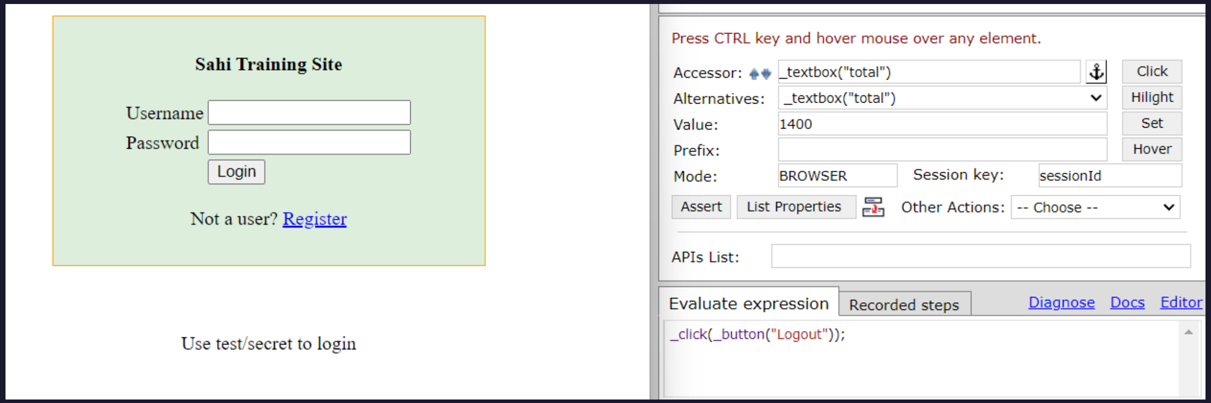
```
_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1450", _getValue(_textbox("total")));
```

```
[Assertion Failed]
Expected:"1450"
Actual:"1400"
```



```
_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1400", _getValue(_textbox("total")));
```

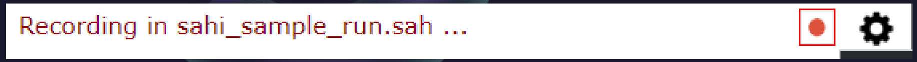
```
Appended:
_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1400", _getValue(_textbox("total")));
```



Press CTRL key and hover mouse over any element.

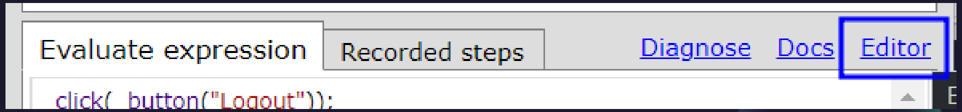
Accessor:	_textbox("total")
Alternatives:	_textbox("total")
Value:	1400
Prefix:	
Mode:	BROWSER
Session key:	sessionId
Other Actions:	-- Choose --
APIs List:	

```
_click(_button("Logout"));
```

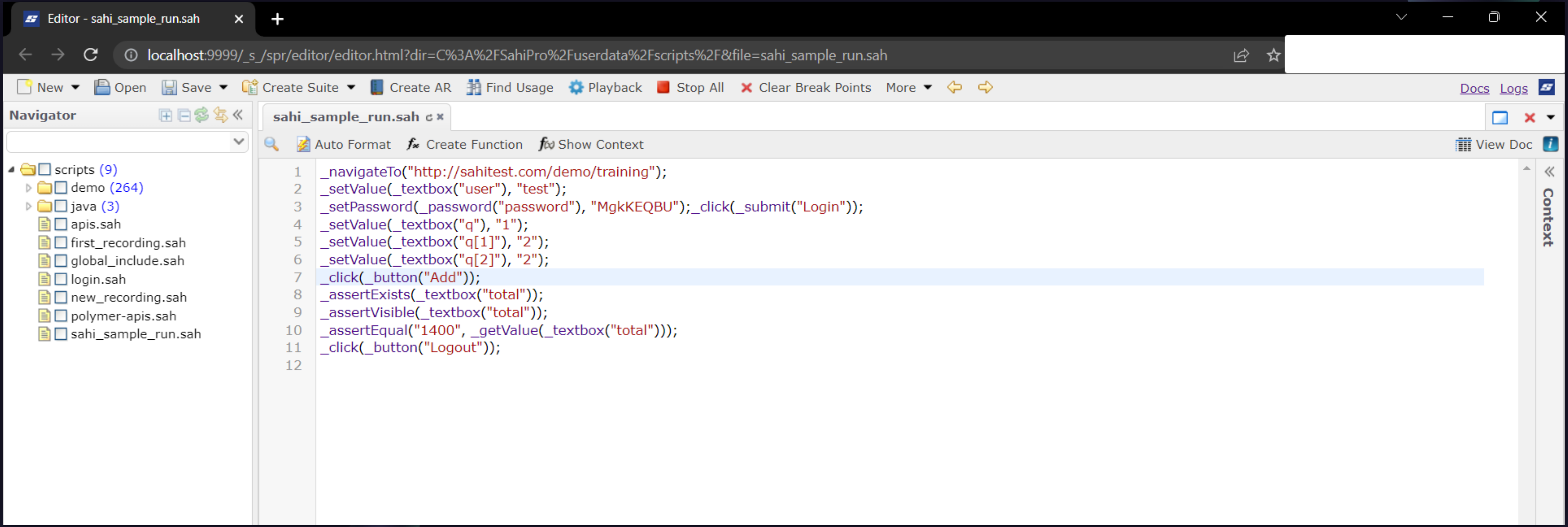


Recording in sahi_sample_run.sah ...

- Click on Editor to open Sahi Scripts Editor. This opens browser and reaches localhost:9999 website to open editor hosted locally by Sahi.



```
_click(_button("Logout"));
```



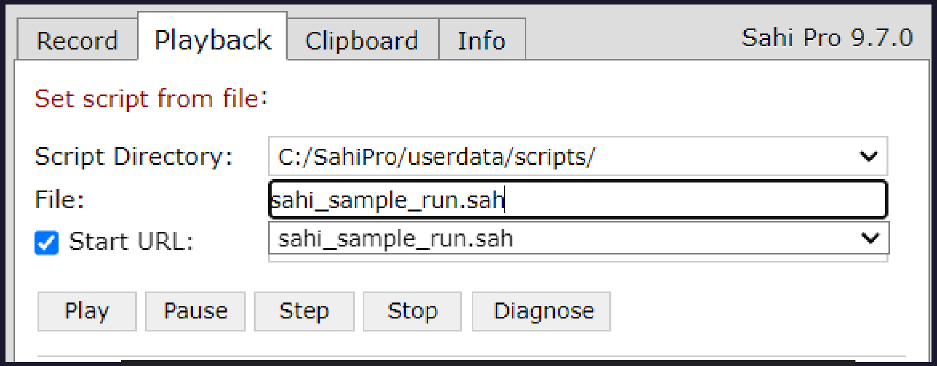
```
_navigateTo("http://sahitest.com/demo/training");
_setValue(_textbox("user"), "test");
_setPassword(_password("password"), "MgkKEQBU");_click(_submit("Login"));
_setValue(_textbox("q"), "1");
_setValue(_textbox("q[1]"), "2");
_setValue(_textbox("q[2]"), "2");
_click(_button("Add"));
_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1400", _getValue(_textbox("total")));
_click(_button("Logout"));
```

- User can see the recorded steps while automating web application. All steps are recorded in JavaScript statements. Main advantage is user does not have to deal with xpaths or CSS selectors to identify an element on DOM. Wait statements are also taken care implicitly by Sahi.

- Let's playback the script that we just recorded. Open Sahi Controller, and go to Playback tab.

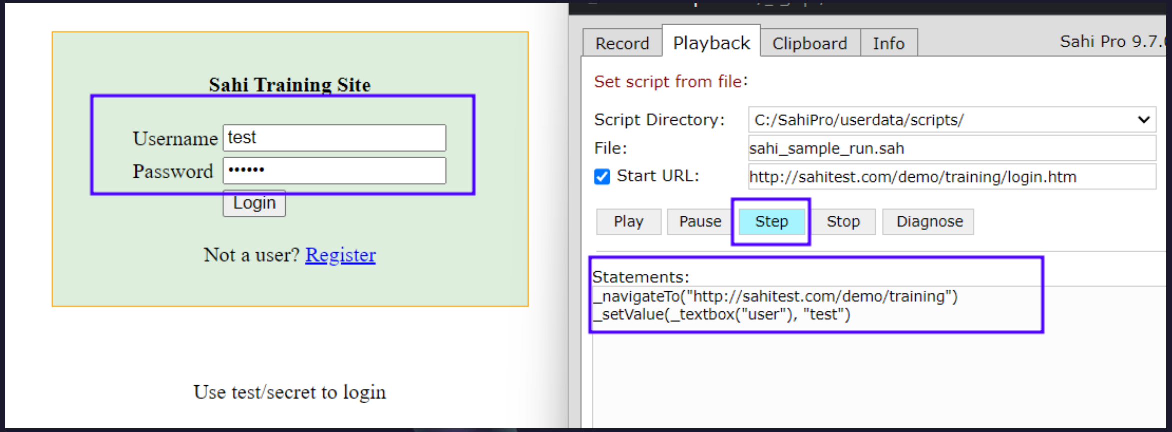
- Clicking Play button starts the script and all Statements executed are shown in Statements window below.

- User can use Pause button to stop the script execution.
- Step button is used to run the script step by step.



Set script from file:

Script Directory: C:/SahiPro/userdata/scripts/
File: sahi_sample_run.sah
☑ Start URL: sahi_sample_run.sah

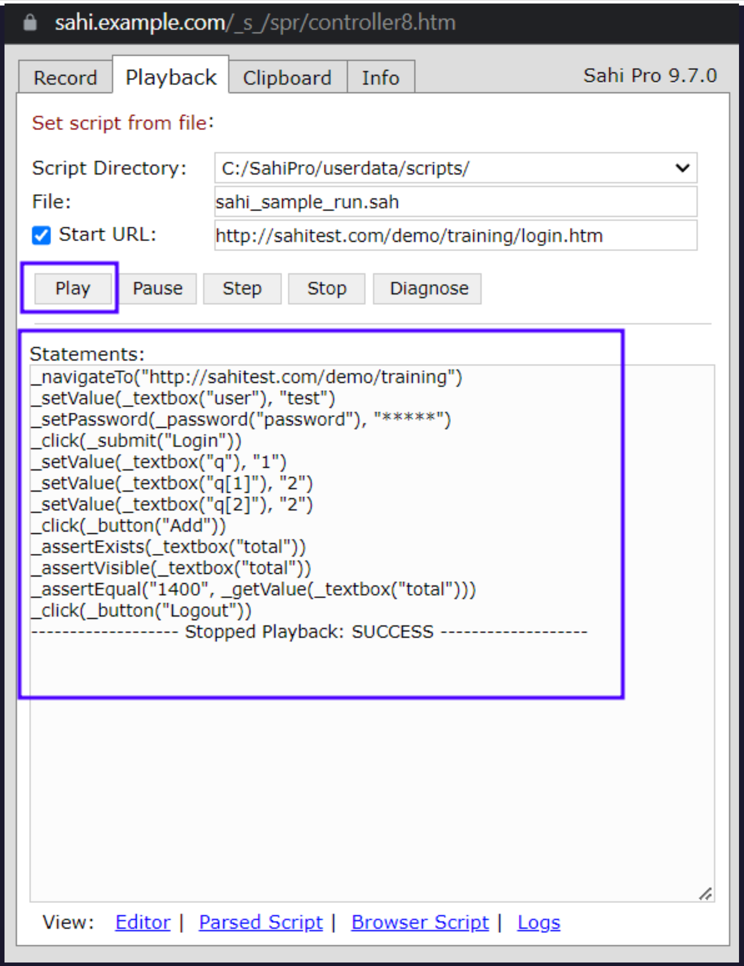


Username: test
Password: ••••••

Set script from file:

Script Directory: C:/SahiPro/userdata/scripts/
File: sahi_sample_run.sah
☑ Start URL: http://sahitest.com/demo/training/login.htm

```
Statements:
_navigateTo("http://sahitest.com/demo/training")
_setValue(_textbox("user"), "test")
```

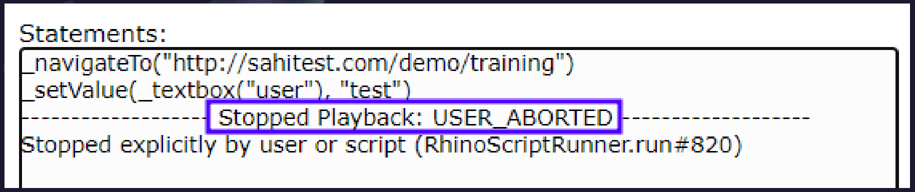

sahi.example.com/_s_/spr/controller8.htm
Record
Playback
Clipboard
Info
Sahi Pro 9.7.0
Play
Pause
Step
Stop
Diagnose
View: Editor | Parsed Script | Browser Script | Logs

Set script from file:

Script Directory: C:/SahiPro/userdata/scripts/
File: sahi_sample_run.sah
☑ Start URL: http://sahitest.com/demo/training/login.htm

```
Statements:
_navigateTo("http://sahitest.com/demo/training")
_setValue(_textbox("user"), "test")
_setPassword(_password("password"), "*****")
_click(_submit("Login"))
_setValue(_textbox("q"), "1")
_setValue(_textbox("q[1]"), "2")
_setValue(_textbox("q[2]"), "2")
_click(_button("Add"))
_assertExists(_textbox("total"))
_assertVisible(_textbox("total"))
_assertEqual("1400", _getValue(_textbox("total")))
_click(_button("Logout"))
------------------ Stopped Playback: SUCCESS ------------------
```

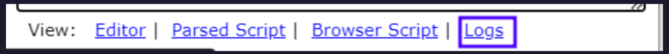
- Stop button brings execution to halt and shows message in Statements tab



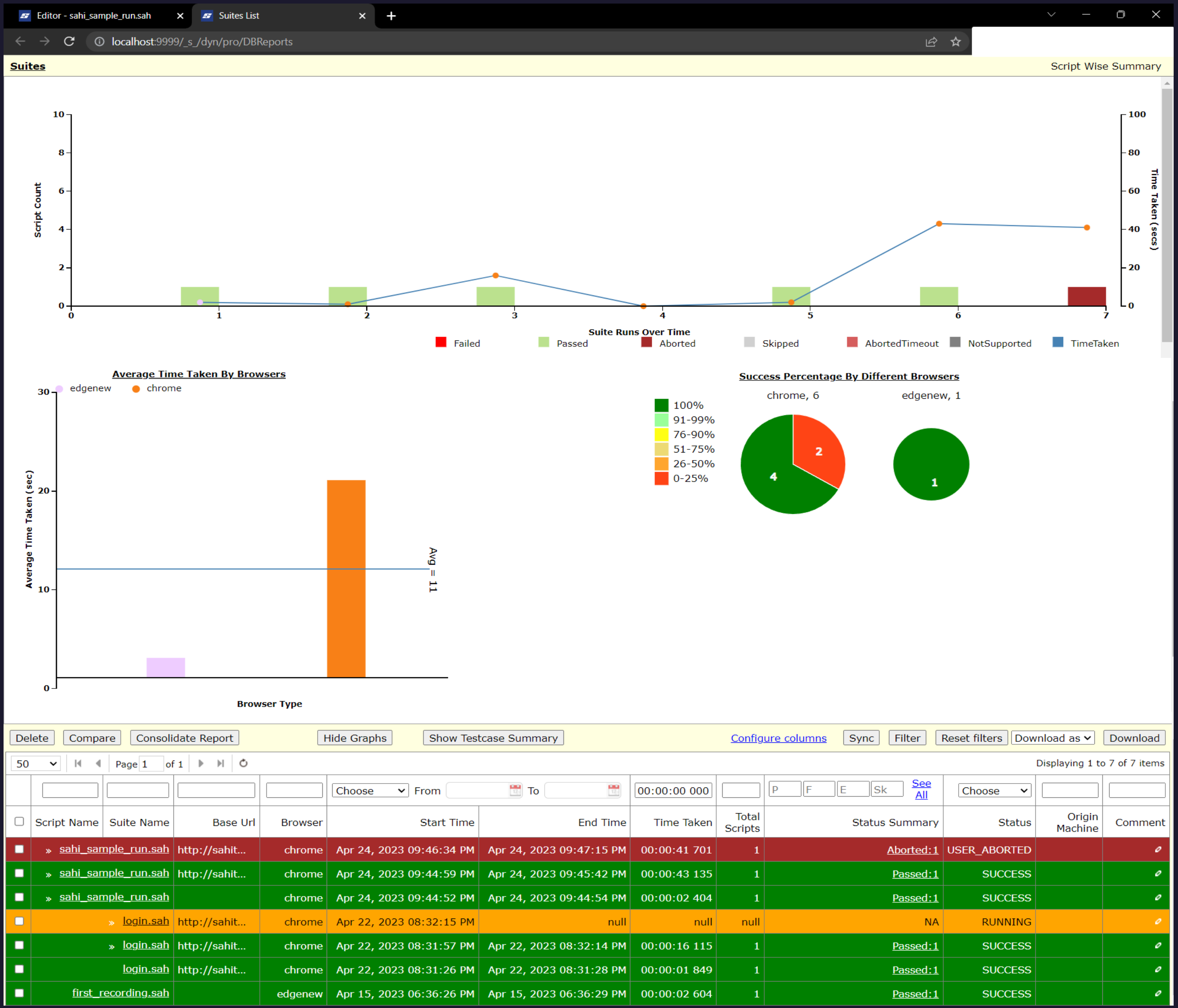
```
Statements:
_navigateTo("http://sahitest.com/demo/training")
_setValue(_textbox("user"), "test")
------------------ Stopped Playback: USER_ABORTED ------------------
Stopped explicitly by user or script (RhinoScriptRunner.run#820)
```

Logging

- User can go through the logs using Logs link in Controller window.


View: Editor | Parsed Script | Browser Script | Logs

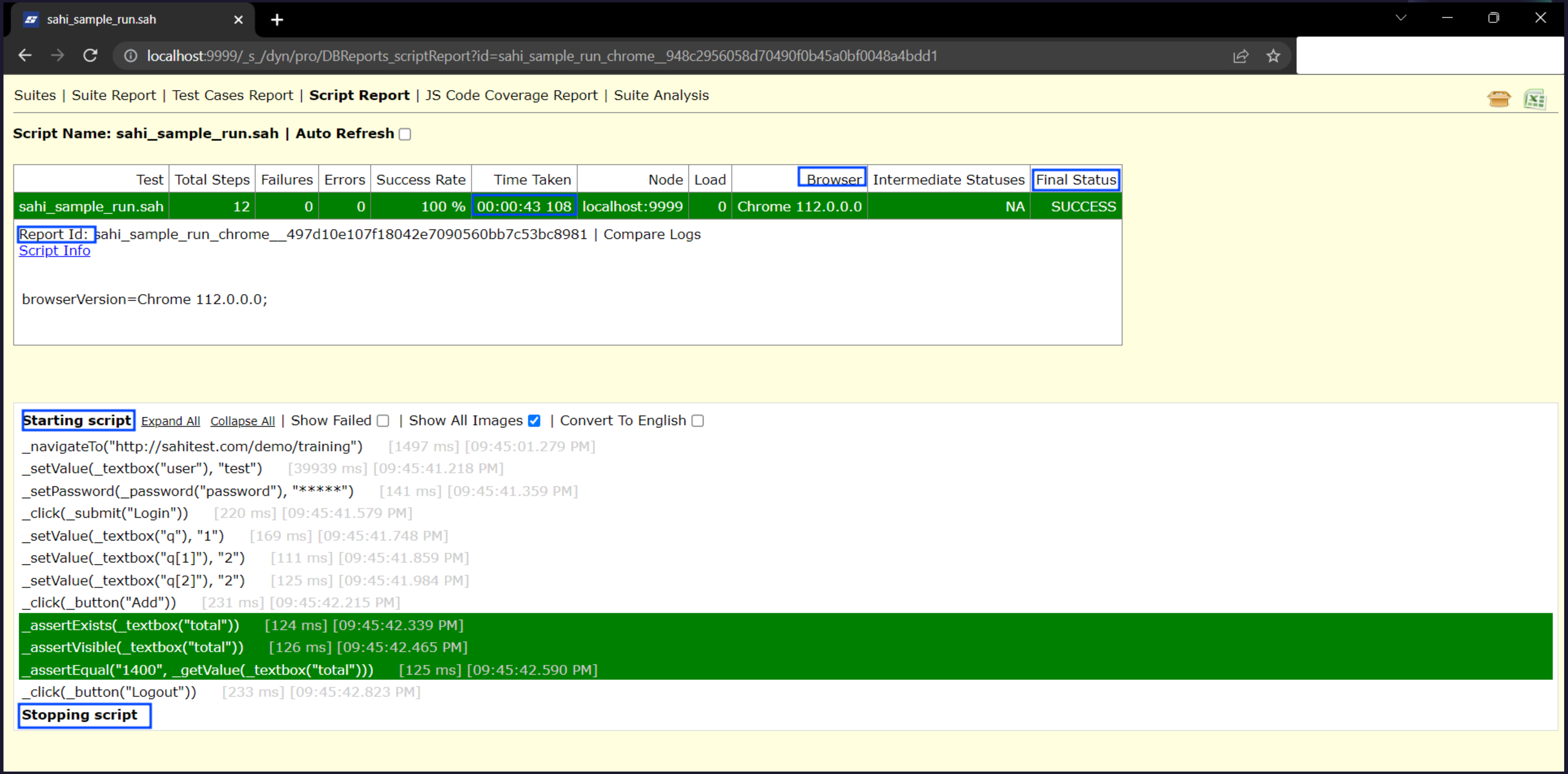
- Logs contains suite runs, Average Time Taken By Browsers graph, Success Percentage By Different Browsers pie chart and all the Playbacks conducted.



Displaying 1 to 7 of 7 items

Script Name	Suite Name	Base Url	Browser	Start Time	End Time	Time Taken	Total Scripts	Status Summary	Status	Origin Machine	Comment
» sahi_sample_run.sah		http://sahit...	chrome	Apr 24, 2023 09:46:34 PM	Apr 24, 2023 09:47:15 PM	00:00:41 701	1	Aborted:1	USER_ABORTED		
» sahi_sample_run.sah		http://sahit...	chrome	Apr 24, 2023 09:44:59 PM	Apr 24, 2023 09:45:42 PM	00:00:43 135	1	Passed:1	SUCCESS		
» sahi_sample_run.sah			chrome	Apr 24, 2023 09:44:52 PM	Apr 24, 2023 09:44:54 PM	00:00:02 404	1	Passed:1	SUCCESS		
» login.sah		http://sahit...	chrome	Apr 22, 2023 08:32:15 PM	null	null	null	NA	RUNNING		
» login.sah		http://sahit...	chrome	Apr 22, 2023 08:31:57 PM	Apr 22, 2023 08:32:14 PM	00:00:16 115	1	Passed:1	SUCCESS		
login.sah		http://sahit...	chrome	Apr 22, 2023 08:31:26 PM	Apr 22, 2023 08:31:28 PM	00:00:01 849	1	Passed:1	SUCCESS		
first_recording.sah			edgenew	Apr 15, 2023 06:36:26 PM	Apr 15, 2023 06:36:29 PM	00:00:02 604	1	Passed:1	SUCCESS		

- User can go through the Sahi reports to know more about the execution process, the time taken for each action to run.



sahi_sample_run.sah

localhost:9999/_s_/dyn/pro/DBReports_scriptReport?id=sahi_sample_run_chrome__948c2956058d70490f0b45a0bf0048a4bdd1

Suites | Suite Report | Test Cases Report | **Script Report** | JS Code Coverage Report | Suite Analysis

Script Name: sahi_sample_run.sah | Auto Refresh ☐

Test	Total Steps	Failures	Errors	Success Rate	Time Taken	Node	Load	Browser	Intermediate Statuses	Final Status
sahi_sample_run.sah	12	0	0	100 %	00:00:43 108	localhost:9999	0	Chrome 112.0.0.0	NA	SUCCESS

Report Id: sahi_sample_run_chrome__497d10e107f18042e7090560bb7c53bc8981 | Compare Logs
Script Info

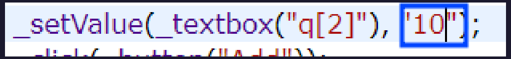
browserVersion=Chrome 112.0.0.0;

Starting script Expand All Collapse All | Show Failed ☐ | Show All Images ☑ | Convert To English ☐

```
_navigateTo("http://sahitest.com/demo/training")   [1497 ms] [09:45:01.279 PM]
_setValue(_textbox("user"), "test")   [39939 ms] [09:45:41.218 PM]
_setPassword(_password("password"), "*****")   [141 ms] [09:45:41.359 PM]
_click(_submit("Login"))   [220 ms] [09:45:41.579 PM]
_setValue(_textbox("q"), "1")   [169 ms] [09:45:41.748 PM]
_setValue(_textbox("q[1]"), "2")   [111 ms] [09:45:41.859 PM]
_setValue(_textbox("q[2]"), "2")   [125 ms] [09:45:41.984 PM]
_click(_button("Add"))   [231 ms] [09:45:42.215 PM]
_assertExists(_textbox("total"))   [124 ms] [09:45:42.339 PM]
_assertVisible(_textbox("total"))   [126 ms] [09:45:42.465 PM]
_assertEqual("1400", _getValue(_textbox("total")))   [125 ms] [09:45:42.590 PM]
_click(_button("Logout"))   [233 ms] [09:45:42.823 PM]
```

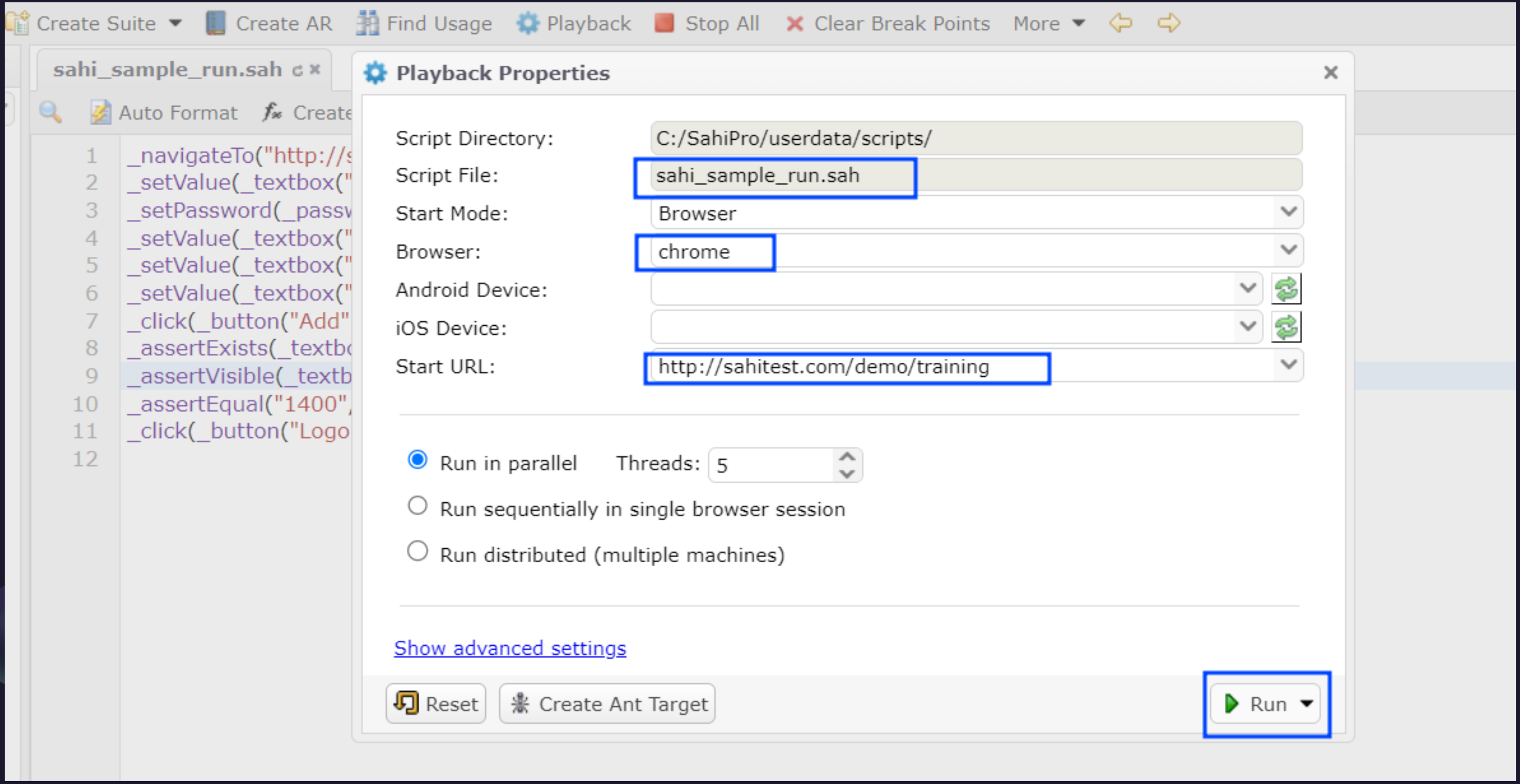
Stopping script

Let's see a failure test case in the script. Changing quantity of book 3 in the cart will give us a different total. When the grand total is not equal to 1400, Sahi logs an error in the test case.



```
_setValue(_textbox("q[2]"), "10");
```

User can playback the script from Sahi Editor without reaching Controller. Select the script file, browser and start URL and click on Run button to start the playback.



```
_navigateTo("http://s
_setValue(_textbox("
_setPassword(_passw
_setValue(_textbox("
_setValue(_textbox("
_setValue(_textbox("
_click(_button("Add"
_assertExists(_textb
_assertVisible(_textb
_assertEqual("1400",
_click(_button("Logo
```

Playback Properties

Script Directory: C:/SahiPro/userdata/scripts/
Script File: sahi_sample_run.sah
Start Mode: Browser
Browser: chrome
Android Device:
iOS Device:
Start URL: http://sahitest.com/demo/training

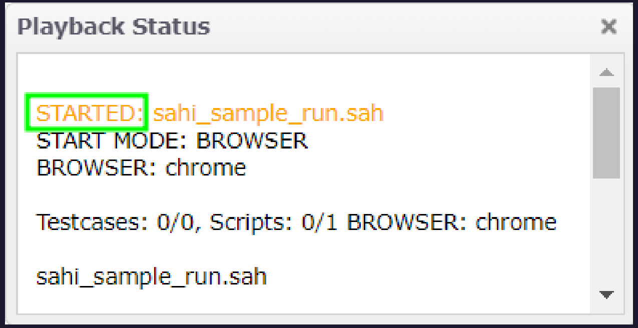
[✓ Run in parallel] Threads: 5
Run sequentially in single browser session
Run distributed (multiple machines)

Let's see a failure test case in the script. Changing quantity of book 3 in the cart will give us a different total. When the grand total is not equal to 1400, Sahi logs an error in the test case.

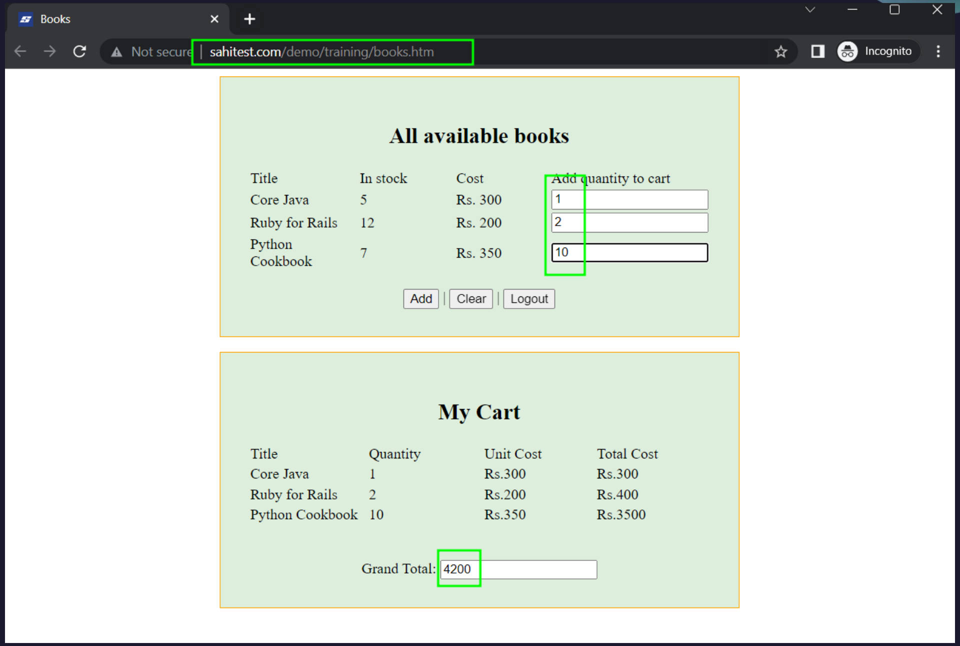
User can playback the script from Sahi Editor without reaching Controller.

Before execution of any step, Sahi waits for any Ajax activity, network activity to subside.

If a step seems to be failing, Sahi waits for 2 seconds and tries to re-execute that step. It will do this 5 times, between the retries, if the system recovers, then Sahi will execute that step, else it'll mark the step as a failure.



```
STARTED: sahi_sample_run.sah
START MODE: BROWSER
BROWSER: chrome

Testcases: 0/0, Scripts: 0/1 BROWSER: chrome

sahi_sample_run.sah
```



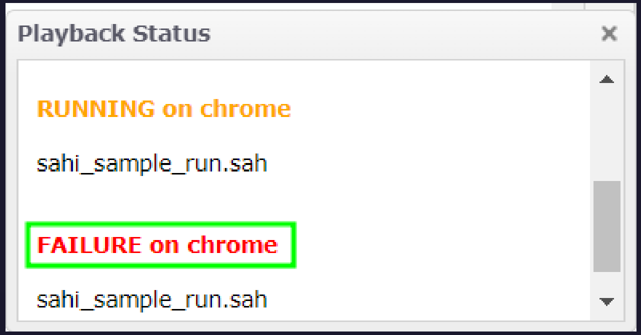
All available books

Title	In stock	Cost	Add quantity to cart
Core Java	5	Rs. 300	1
Ruby for Rails	12	Rs. 200	2
Python Cookbook	7	Rs. 350	10

My Cart

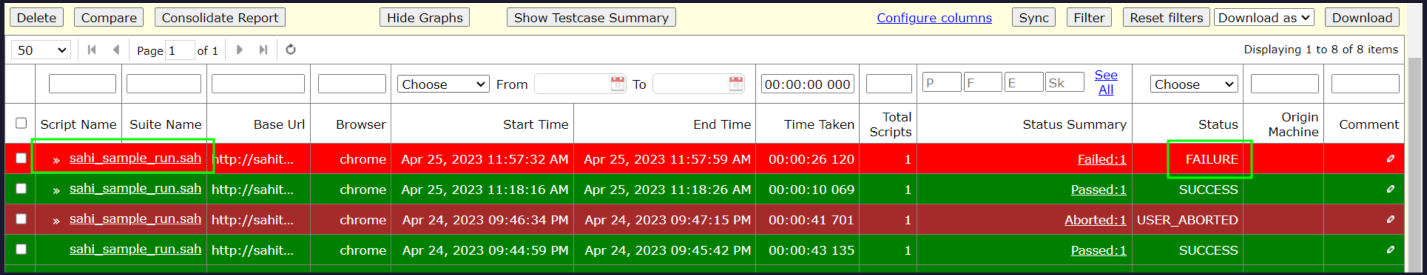
Title	Quantity	Unit Cost	Total Cost
Core Java	1	Rs.300	Rs.300
Ruby for Rails	2	Rs.200	Rs.400
Python Cookbook	10	Rs.350	Rs.3500

Grand Total: 4200



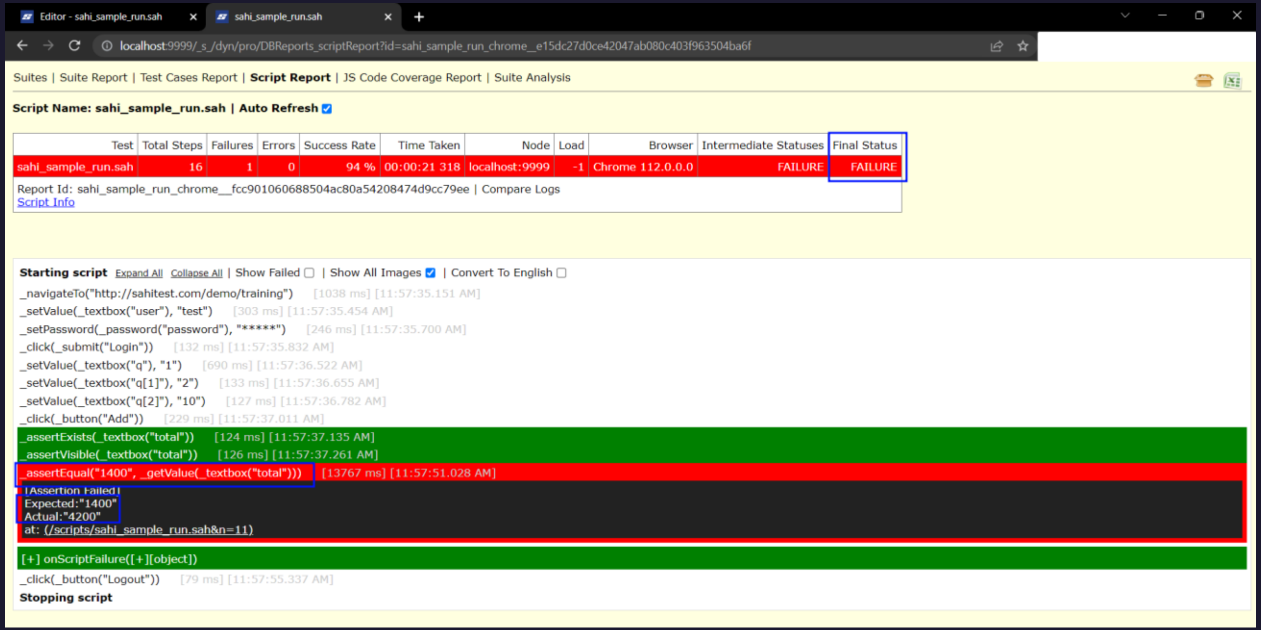
```
RUNNING on chrome

sahi_sample_run.sah

FAILURE on chrome

sahi_sample_run.sah
```

Let’s go through the logs, we see that script itself is in red



Script Name	Suite Name	Base Url	Browser	Start Time	End Time	Time Taken	Total Scripts	Status Summary	Status	Origin Machine	Comment
» sahi_sample_run.sah		http://sahit...	chrome	Apr 25, 2023 11:57:32 AM	Apr 25, 2023 11:57:59 AM	00:00:26 120	1	Failed:1	FAILURE		
» sahi_sample_run.sah		http://sahit...	chrome	Apr 25, 2023 11:18:16 AM	Apr 25, 2023 11:18:26 AM	00:00:10 069	1	Passed:1	SUCCESS		
» sahi_sample_run.sah		http://sahit...	chrome	Apr 24, 2023 09:46:34 PM	Apr 24, 2023 09:47:15 PM	00:00:41 701	1	Aborted:1	USER_ABORTED		
sahi_sample_run.sah		http://sahit...	chrome	Apr 24, 2023 09:44:59 PM	Apr 24, 2023 09:45:42 PM	00:00:43 135	1	Passed:1	SUCCESS		

Opening the script, the assert step is marked in red and showing the expected and actual values mismatch. Opening onScritpFailure step shows the screenshot when the assert statement failure happened during test run.


Editor - sahi_sample_run.sah
sahi_sample_run.sah
localhost:9999/_s_/dyn/pro/DBReports_scriptReport?id=sahi_sample_run_chrome__e15dc27d0ce42047ab080c403f963504ba6f
Suites | Suite Report | Test Cases Report | Script Report | JS Code Coverage Report | Suite Analysis

Script Name: sahi_sample_run.sah | Auto Refresh ☑

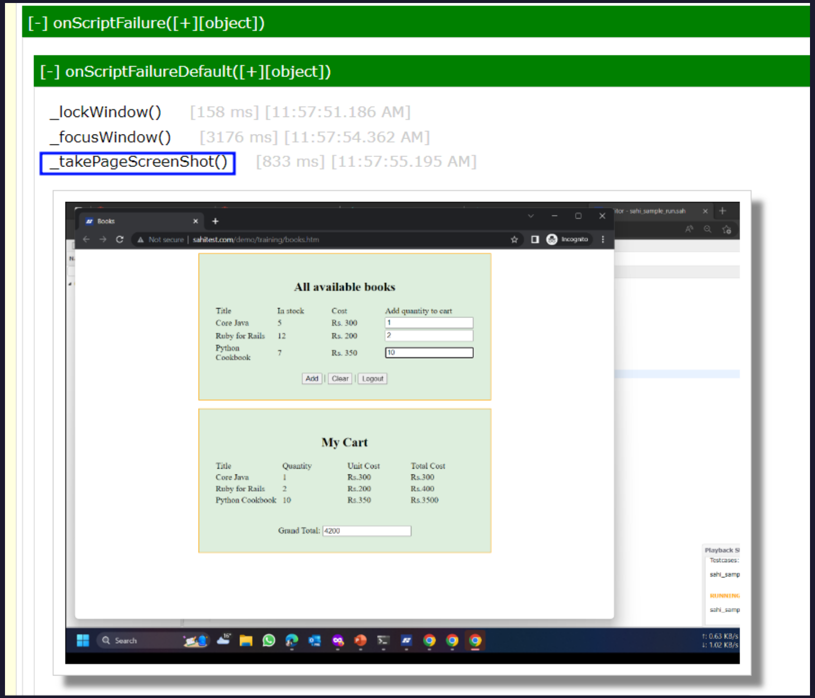
Test	Total Steps	Failures	Errors	Success Rate	Time Taken	Node	Load	Browser	Intermediate Statuses	Final Status
sahi_sample_run.sah	16	1	0	94 %	00:00:21 318	localhost:9999	-1	Chrome 112.0.0.0	FAILURE	FAILURE

Report Id: sahi_sample_run_chrome__fcc901060688504ac80a54208474d9cc79ee | Compare Logs
Script Info

Starting script Expand All Collapse All | Show Failed ☐ | Show All Images ☑ | Convert To English ☐

```
_navigateTo("http://sahitest.com/demo/training")   [1038 ms] [11:57:35.151 AM]
_setValue(_textbox("user"), "test")   [303 ms] [11:57:35.454 AM]
_setPassword(_password("password"), "*****")   [246 ms] [11:57:35.700 AM]
_click(_submit("Login"))   [132 ms] [11:57:35.832 AM]
_setValue(_textbox("q"), "1")   [690 ms] [11:57:36.522 AM]
_setValue(_textbox("q[1]"), "2")   [133 ms] [11:57:36.655 AM]
_setValue(_textbox("q[2]"), "10")   [127 ms] [11:57:36.782 AM]
_click(_button("Add"))   [229 ms] [11:57:37.011 AM]
_assertExists(_textbox("total"))   [124 ms] [11:57:37.135 AM]
_assertVisible(_textbox("total"))   [126 ms] [11:57:37.261 AM]
_assertEqual("1400", _getValue(_textbox("total")))   [13767 ms] [11:57:51.028 AM]
 [Assertion Failed]
 Expected:"1400"
 Actual:"4200"
 at: (/scripts/sahi_sample_run.sah&n=11)
[+] onScriptFailure([+][object])
_click(_button("Logout"))   [79 ms] [11:57:55.337 AM]
```

Stopping script



```
[-] onScriptFailure([+][object])
[-] onScriptFailureDefault([+][object])
_lockWindow()   [158 ms] [11:57:51.186 AM]
_focusWindow()   [3176 ms] [11:57:54.362 AM]
_takePageScreenShot()   [833 ms] [11:57:55.195 AM]
```

All available books

Title	In stock	Cost	Add quantity to cart
Core Java	5	Rs. 300	1
Ruby for Rails	12	Rs. 200	2
Python Cookbook	7	Rs. 350	10

Add | Clear | Logout

My Cart

Title	Quantity	Unit Cost	Total Cost
Core Java	1	Rs.300	Rs.300
Ruby for Rails	2	Rs.200	Rs.400
Python Cookbook	10	Rs.350	Rs.3500

Grand Total: 4200

Library Functions

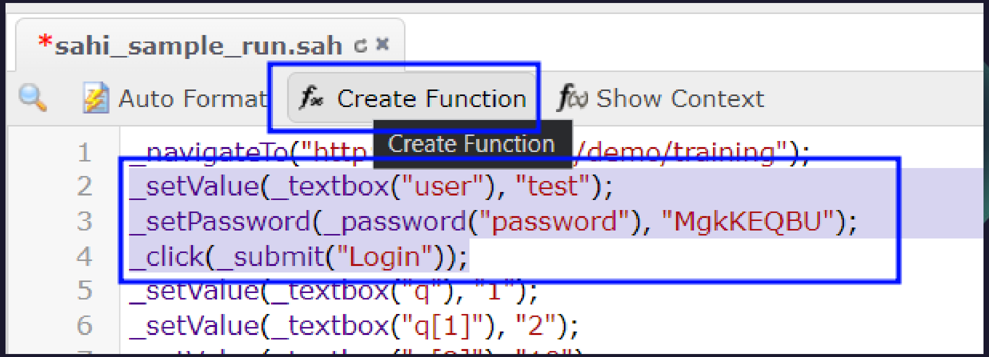
The steps that we see in the Sahi Script are low level instructions to the computer to perform actions on a browser.

Language of the business is much more in human speaking terms like – login to the system, add books quantities to the cart, check the total amount, and logout from the system.

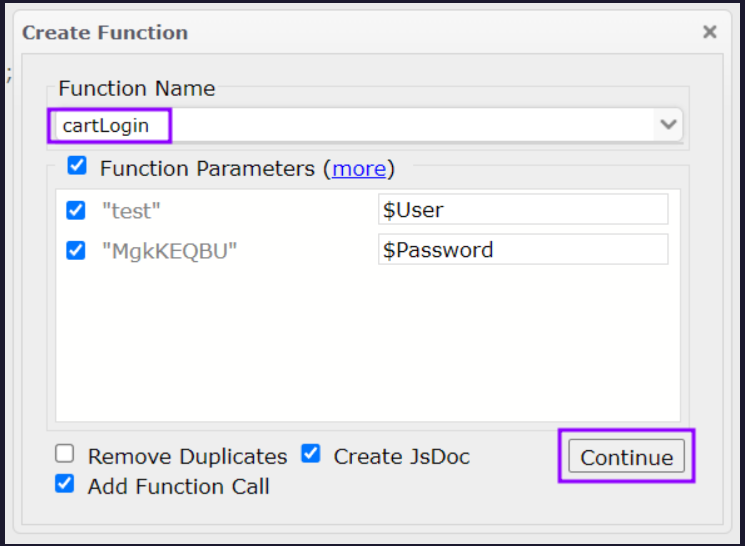
So, we need to create Business Level Abstractions out of the steps that we recorded.

We create Business Level Abstractions using Functions in Sahi.

To create a function, we select steps that pertain to a logical business step and click on Create Function button, provide a function name and click on continue.



```
_navigateTo("http                 /demo/training");
_setValue(_textbox("user"), "test");
_setPassword(_password("password"), "MgkKEQBU");
_click(_submit("Login"));
_setValue(_textbox("q"), "1");
_setValue(_textbox("q[1]"), "2");
```

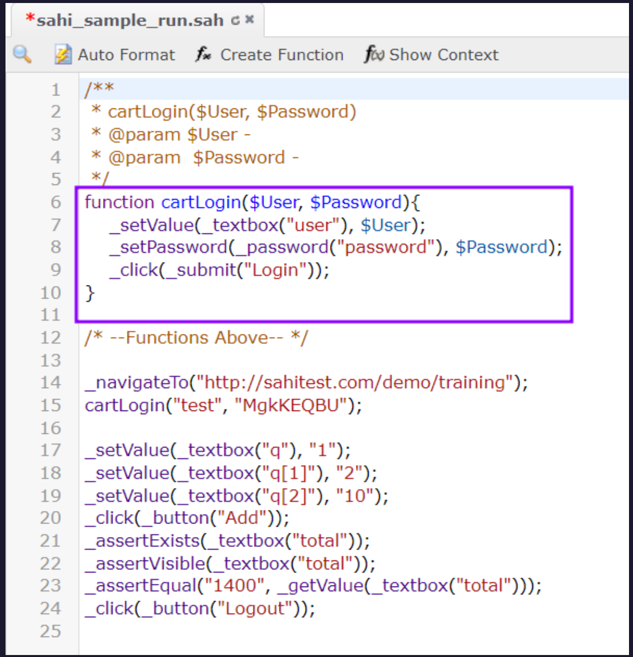


Function Name: cartLogin

[✓] Function Parameters (more)

[✓] "test"	$User
[✓] "MgkKEQBU"	$Password

[ ] Remove Duplicates [✓] Create JsDoc
[✓] Add Function Call

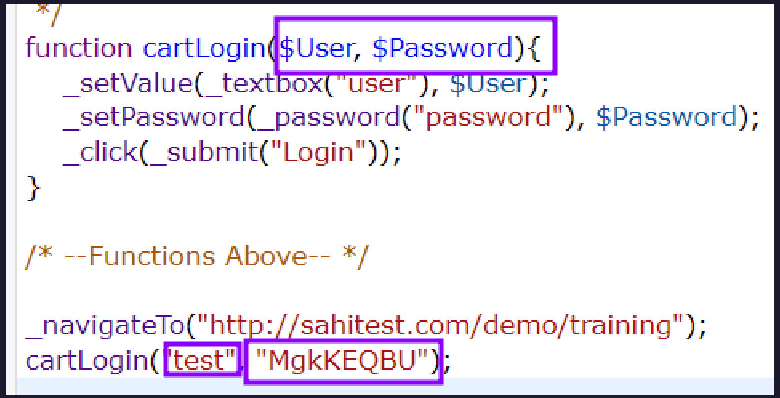


```
/**
 * cartLogin($User, $Password)
 * @param $User -
 * @param  $Password -
 */
function cartLogin($User, $Password){
    _setValue(_textbox("user"), $User);
    _setPassword(_password("password"), $Password);
    _click(_submit("Login"));
}

/* --Functions Above-- */

_navigateTo("http://sahitest.com/demo/training");
cartLogin("test", "MgkKEQBU");

_setValue(_textbox("q"), "1");
_setValue(_textbox("q[1]"), "2");
_setValue(_textbox("q[2]"), "10");
_click(_button("Add"));
_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1400", _getValue(_textbox("total")));
_click(_button("Logout"));
```

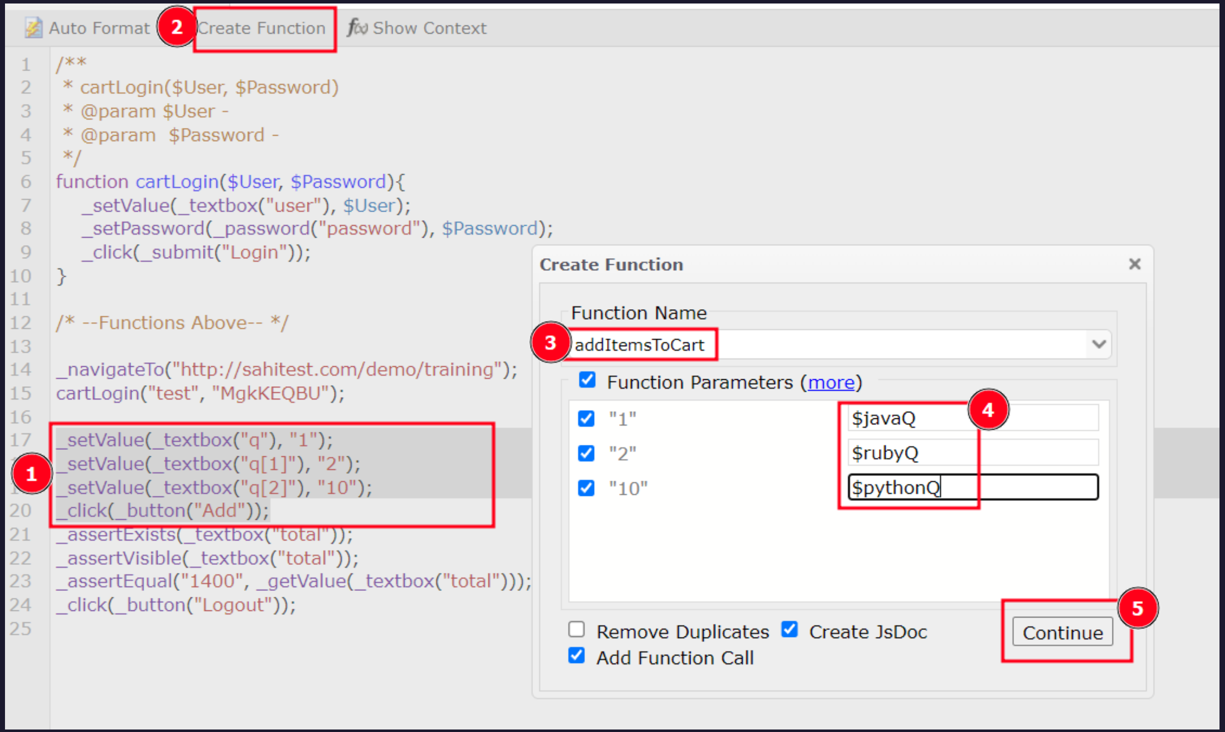
Sahi creates a function and extracts possible parameters from the recorded steps and takes them as arguments to the function.

In the recorded steps place, it will call the created function with the values from the steps.

We do similar procedure for remaining steps, combine the quantity adding steps to a single function.



```
*/
function cartLogin($User, $Password){
    _setValue(_textbox("user"), $User);
    _setPassword(_password("password"), $Password);
    _click(_submit("Login"));
}

/* --Functions Above-- */

_navigateTo("http://sahitest.com/demo/training");
cartLogin("test", "MgkKEQBU");
```



```
/**
 * cartLogin($User, $Password)
 * @param $User -
 * @param  $Password -
 */
function cartLogin($User, $Password){
   _setValue(_textbox("user"), $User);
   _setPassword(_password("password"), $Password);
   _click(_submit("Login"));
}

/* --Functions Above-- */

_navigateTo("http://sahitest.com/demo/training");
cartLogin("test", "MgkKEQBU");

_setValue(_textbox("q"), "1");
_setValue(_textbox("q[1]"), "2");
_setValue(_textbox("q[2]"), "10");
_click(_button("Add"));
_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1400", _getValue(_textbox("total")));
_click(_button("Logout"));
```

Create Function

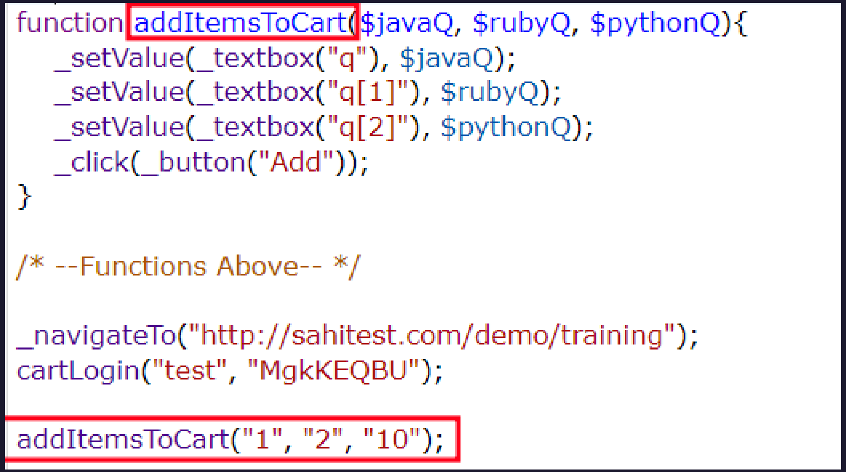
Function Name: addItemsToCart

[✓] Function Parameters (more)

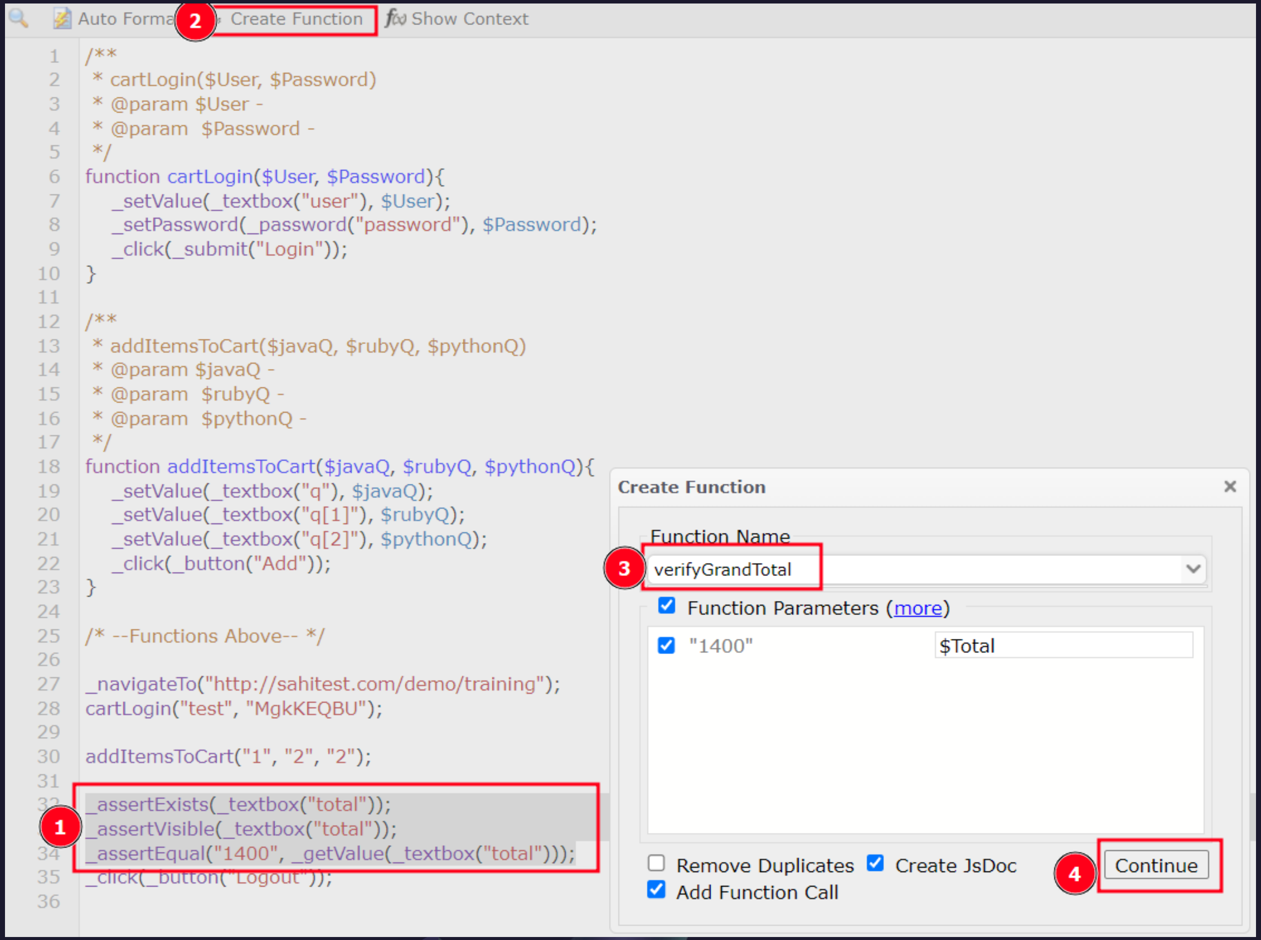
[✓] "1"	$javaQ
[✓] "2"	$rubyQ
[✓] "10"	$pythonQ

[ ] Remove Duplicates [✓] Create JsDoc
[✓] Add Function Call

Continue



```
function addItemsToCart($javaQ, $rubyQ, $pythonQ){
   _setValue(_textbox("q"), $javaQ);
   _setValue(_textbox("q[1]"), $rubyQ);
   _setValue(_textbox("q[2]"), $pythonQ);
   _click(_button("Add"));
}

/* --Functions Above-- */

_navigateTo("http://sahitest.com/demo/training");
cartLogin("test", "MgkKEQBU");

addItemsToCart("1", "2", "10");
```



```
/**
 * cartLogin($User, $Password)
 * @param $User -
 * @param  $Password -
 */
function cartLogin($User, $Password){
    _setValue(_textbox("user"), $User);
    _setPassword(_password("password"), $Password);
    _click(_submit("Login"));
}

/**
 * addItemsToCart($javaQ, $rubyQ, $pythonQ)
 * @param $javaQ -
 * @param  $rubyQ -
 * @param  $pythonQ -
 */
function addItemsToCart($javaQ, $rubyQ, $pythonQ){
    _setValue(_textbox("q"), $javaQ);
    _setValue(_textbox("q[1]"), $rubyQ);
    _setValue(_textbox("q[2]"), $pythonQ);
    _click(_button("Add"));
}

/* --Functions Above-- */

_navigateTo("http://sahitest.com/demo/training");
cartLogin("test", "MgkKEQBU");

addItemsToCart("1", "2", "2");

_assertExists(_textbox("total"));
_assertVisible(_textbox("total"));
_assertEqual("1400", _getValue(_textbox("total")));
_click(_button("Logout"));
```

Create Function

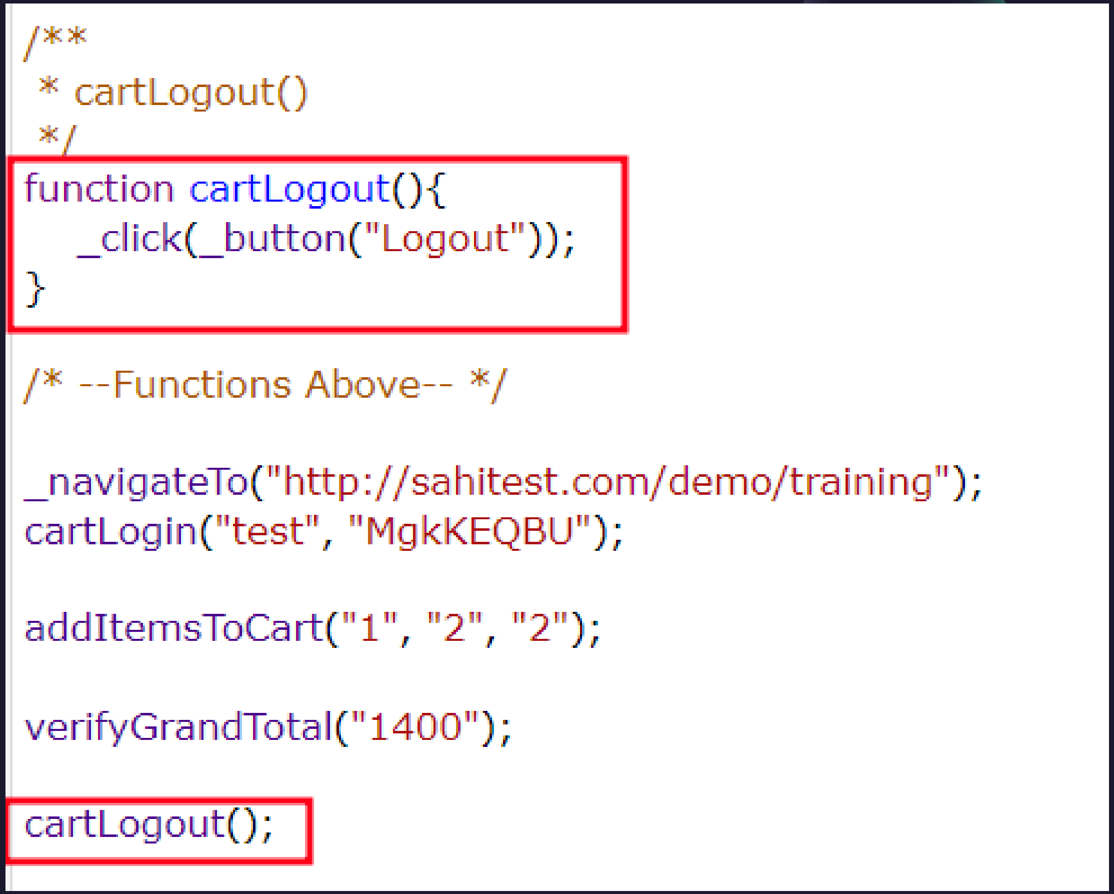
Function Name: verifyGrandTotal

[✓] Function Parameters (more)

[✓] "1400"	$Total

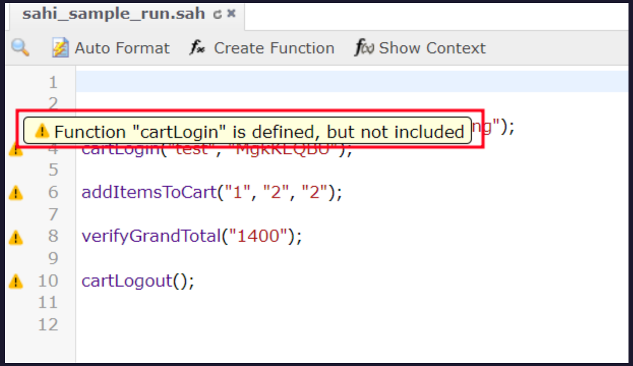
[ ] Remove Duplicates [✓] Create JsDoc
[✓] Add Function Call

Continue



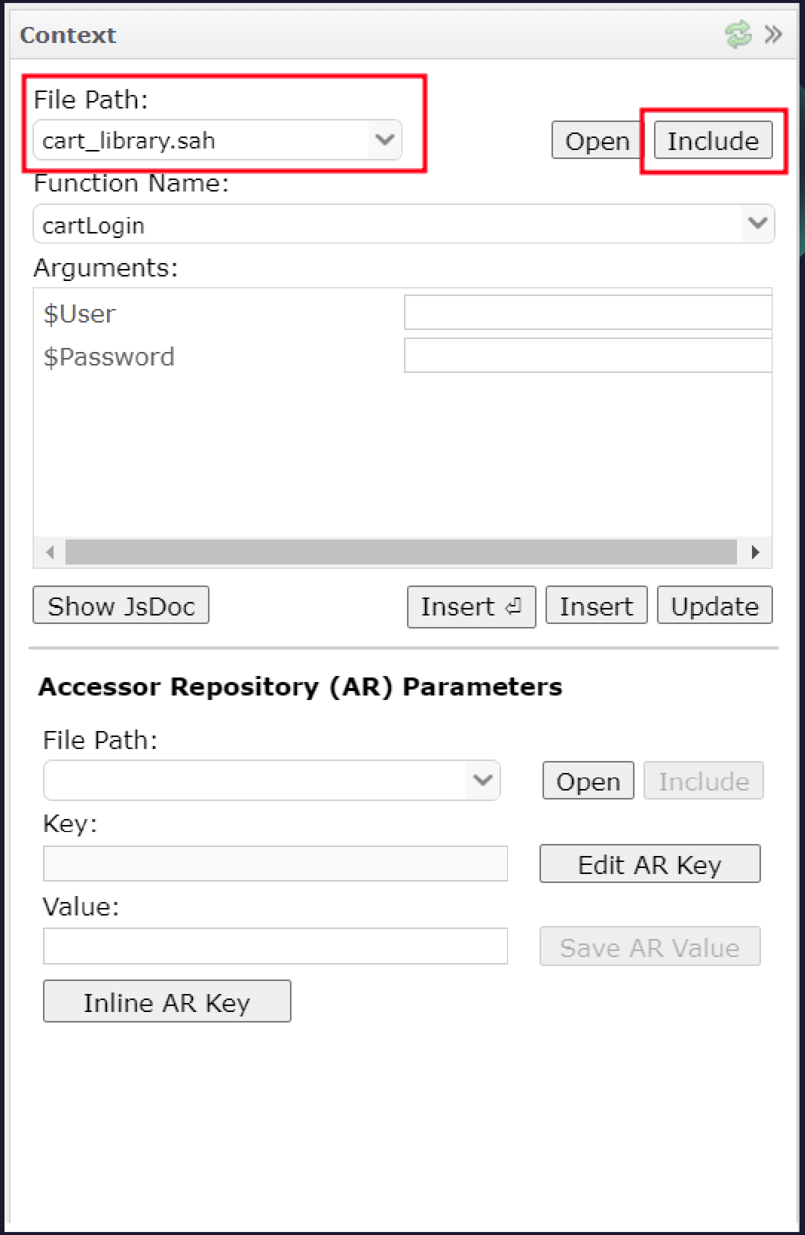
```
/**
 * cartLogout()
 */
function cartLogout(){
    _click(_button("Logout"));
}

/* --Functions Above-- */

_navigateTo("http://sahitest.com/demo/training");
cartLogin("test", "MgkKEQBU");

addItemsToCart("1", "2", "2");

verifyGrandTotal("1400");

cartLogout();
```

To reuse the functions generated, we place them in a separate .sah script file.
Press Control and click on the function to open Context panel.



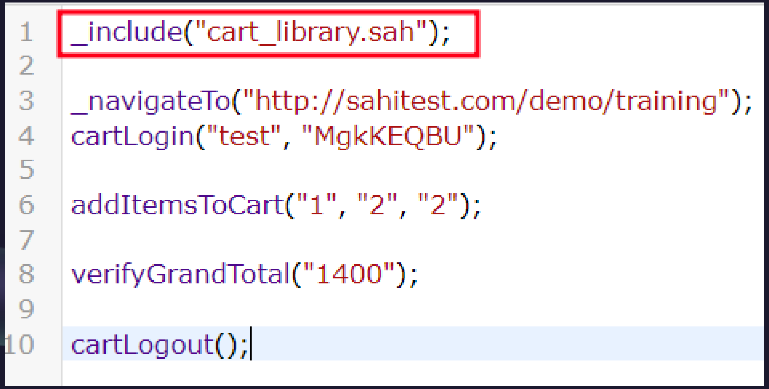
Function "cartLogin" is defined, but not included

```
ng");
cartLogin("test", "MgkKEQBU");

addItemsToCart("1", "2", "2");

verifyGrandTotal("1400");

cartLogout();
```



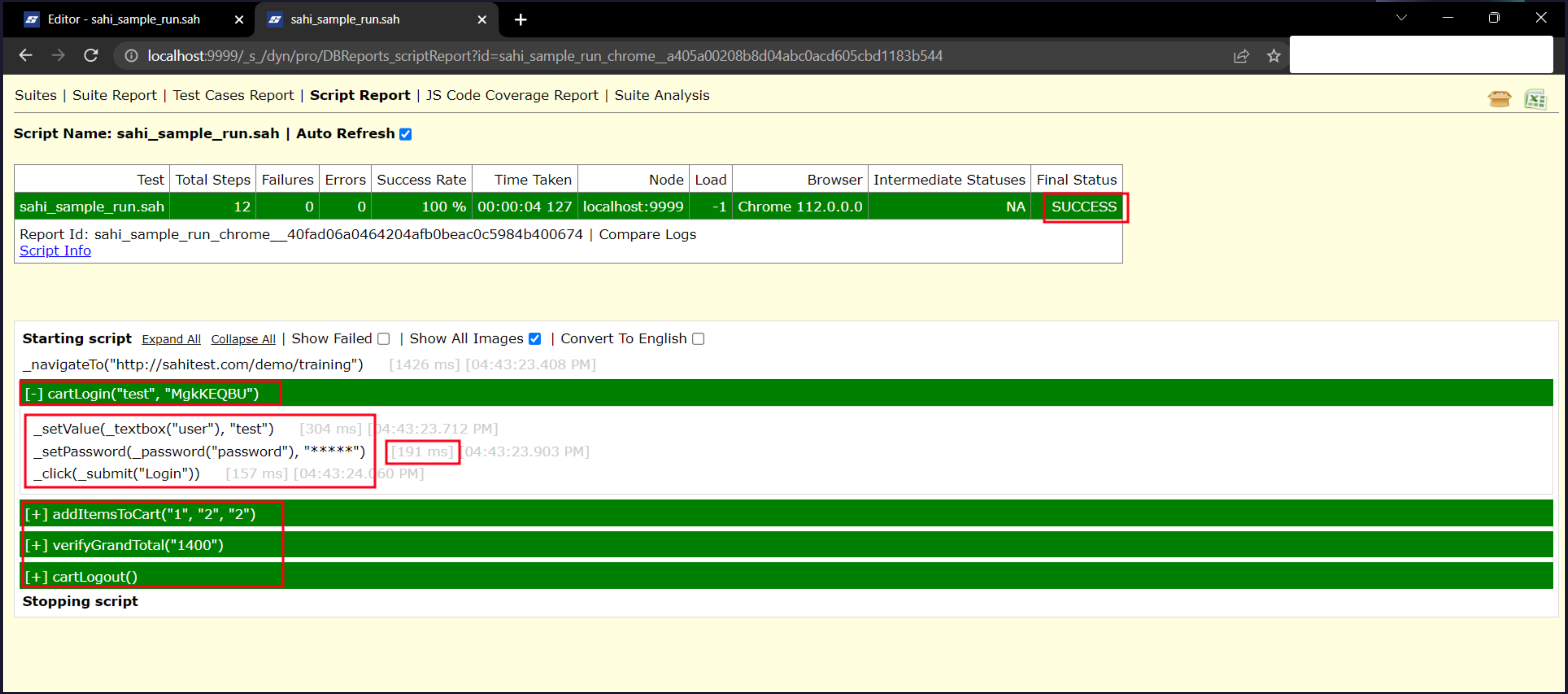
File Path:	cart_library.sah
Function Name:	cartLogin
Arguments:	$User
	$Password

Select the library file from File Path where the function was defined and implemented. Click on Include button to add import statement to the current script.



```
_include("cart_library.sah");

_navigateTo("http://sahitest.com/demo/training");
cartLogin("test", "MgkKEQBU");

addItemsToCart("1", "2", "2");

verifyGrandTotal("1400");

cartLogout();
```

Playback the script.
Let's look at the logs for the execution completed in Sahi Logs window.
Each function is listed in log and time for execution is listed after each step.



Editor - sahi_sample_run.sah
sahi_sample_run.sah
localhost:9999/_s_/dyn/pro/DBReports_scriptReport?id=sahi_sample_run_chrome__a405a00208b8d04abc0acd605cbd1183b544

Suites | Suite Report | Test Cases Report | **Script Report** | JS Code Coverage Report | Suite Analysis

Script Name: sahi_sample_run.sah | Auto Refresh ☑

Test	Total Steps	Failures	Errors	Success Rate	Time Taken	Node	Load	Browser	Intermediate Statuses	Final Status
sahi_sample_run.sah	12	0	0	100 %	00:00:04 127	localhost:9999	-1	Chrome 112.0.0.0	NA	SUCCESS

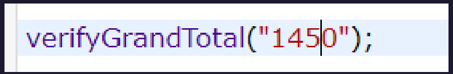
Report Id: sahi_sample_run_chrome__40fad06a0464204afb0beac0c5984b400674 | Compare Logs
Script Info

Starting script Expand All Collapse All | Show Failed ☐ | Show All Images ☑ | Convert To English ☐

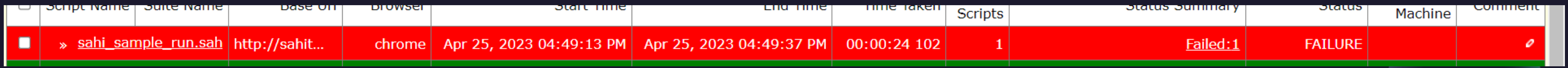
```
_navigateTo("http://sahitest.com/demo/training")   [1426 ms] [04:43:23.408 PM]
[-] cartLogin("test", "MgkKEQBU")
  _setValue(_textbox("user"), "test")   [304 ms] [04:43:23.712 PM]
  _setPassword(_password("password"), "*****")   [191 ms] [04:43:23.903 PM]
  _click(_submit("Login"))   [157 ms] [04:43:24.060 PM]
[+] addItemsToCart("1", "2", "2")
[+] verifyGrandTotal("1400")
[+] cartLogout()
```

Stopping script

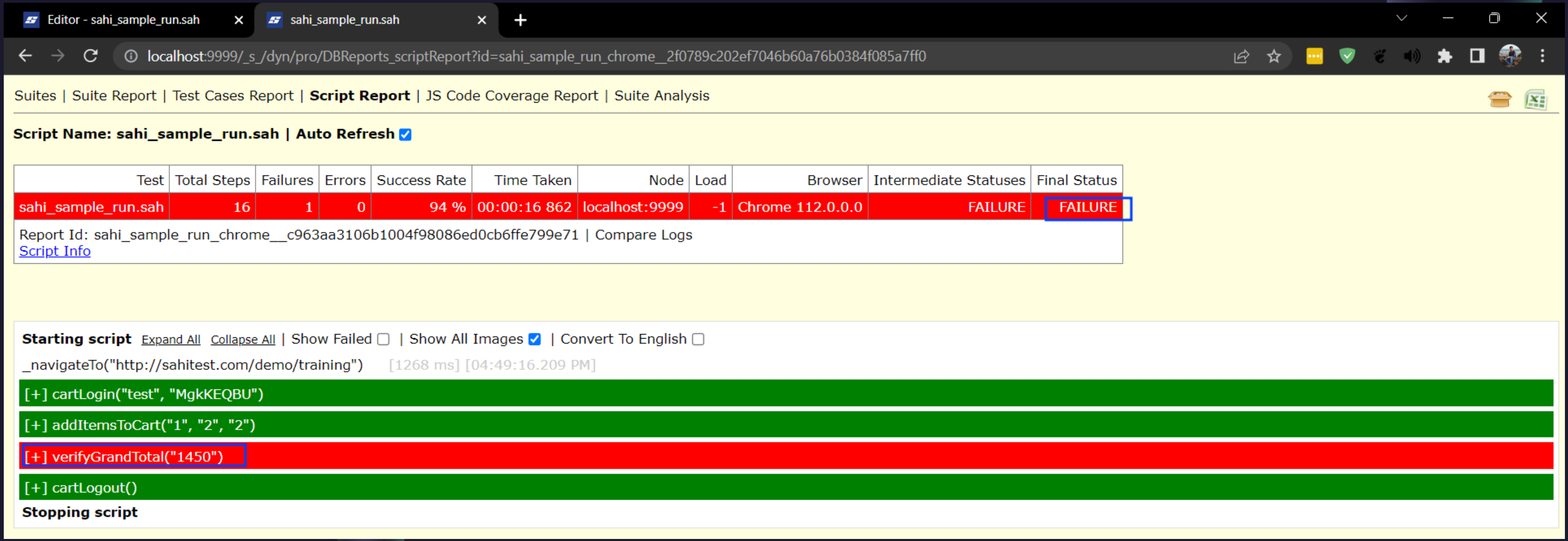
Let's change the script little bit to raise an error in grand total assertion, and see the logs



```
verifyGrandTotal("1450");
```



	Script Name	Suite Name	Base Url	Browser	Start Time	End Time	Time Taken	Scripts	Status Summary	Status	Machine	Comment
☐	» sahi_sample_run.sah		http://sahit...	chrome	Apr 25, 2023 04:49:13 PM	Apr 25, 2023 04:49:37 PM	00:00:24 102	1	Failed:1	FAILURE		



Editor - sahi_sample_run.sah | sahi_sample_run.sah

localhost:9999/_s_/dyn/pro/DBReports_scriptReport?id=sahi_sample_run_chrome__2f0789c202ef7046b60a76b0384f085a7ff0

Suites | Suite Report | Test Cases Report | **Script Report** | JS Code Coverage Report | Suite Analysis

Script Name: sahi_sample_run.sah | Auto Refresh ☑

Test	Total Steps	Failures	Errors	Success Rate	Time Taken	Node	Load	Browser	Intermediate Statuses	Final Status
sahi_sample_run.sah	16	1	0	94 %	00:00:16 862	localhost:9999	-1	Chrome 112.0.0.0	FAILURE	FAILURE

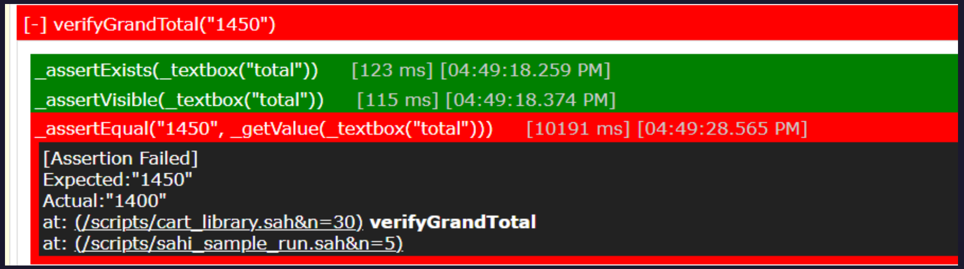
Report Id: sahi_sample_run_chrome__c963aa3106b1004f98086ed0cb6ffe799e71 | Compare Logs
Script Info

Starting script Expand All Collapse All | Show Failed ☐ | Show All Images ☑ | Convert To English ☐

```
_navigateTo("http://sahitest.com/demo/training")   [1268 ms] [04:49:16.209 PM]
[+] cartLogin("test", "MgkKEQBU")
[+] addItemsToCart("1", "2", "2")
[+] verifyGrandTotal("1450")
[+] cartLogout()
```

Stopping script

Grouping as functions let's us detect the code where the assertion failed. Screenshot is also provided in logs for the failure occurred.



```
[-] verifyGrandTotal("1450")
  _assertExists(_textbox("total"))    [123 ms] [04:49:18.259 PM]
  _assertVisible(_textbox("total"))    [115 ms] [04:49:18.374 PM]
  _assertEqual("1450", _getValue(_textbox("total")))    [10191 ms] [04:49:28.565 PM]
  [Assertion Failed]
  Expected:"1450"
  Actual:"1400"
  at: (/scripts/cart_library.sah&n=30) verifyGrandTotal
  at: (/scripts/sahi_sample_run.sah&n=5)
```

Scenarios

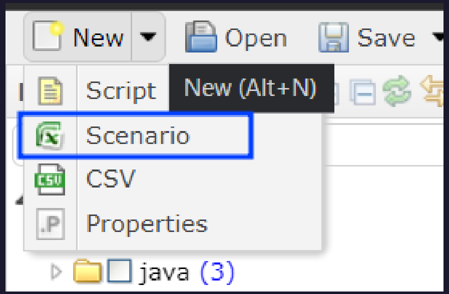
Sahi Scripts are written in JavaScript language.

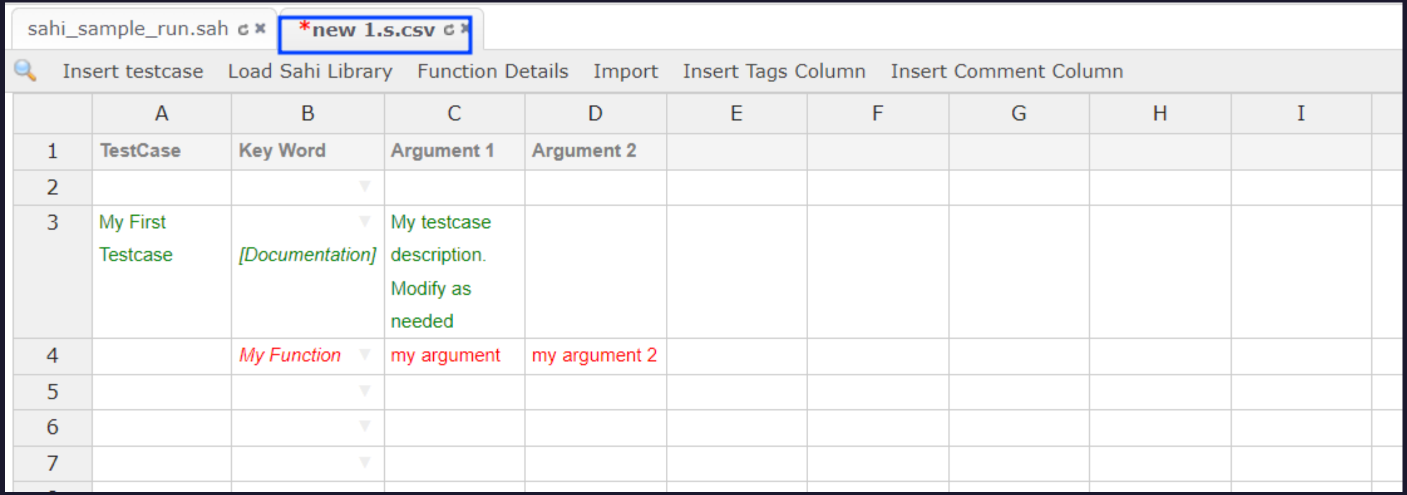
Testers are present where they understand the business functionality of the application, but do not necessarily understand the syntax and semantics of Sahi Script.

To help them participate in the automation process, we have an alternative way of defining our test cases and scenarios. We do that using a scenario file.

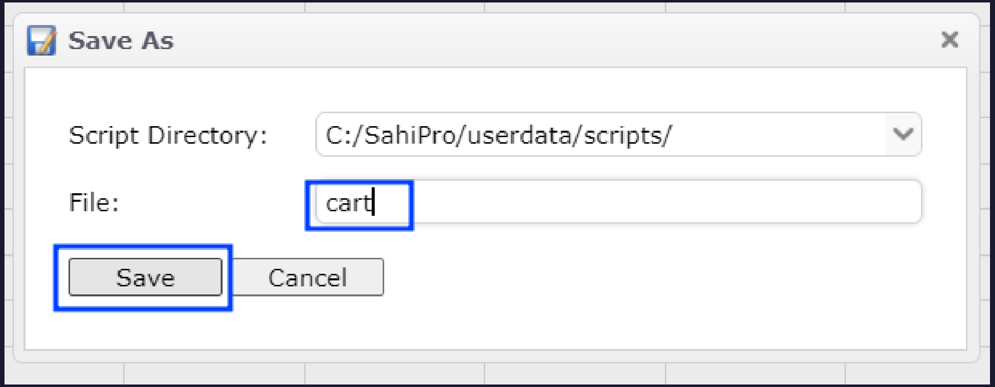
Click on New button and select Scenario button.

Click on Save button and save the file.





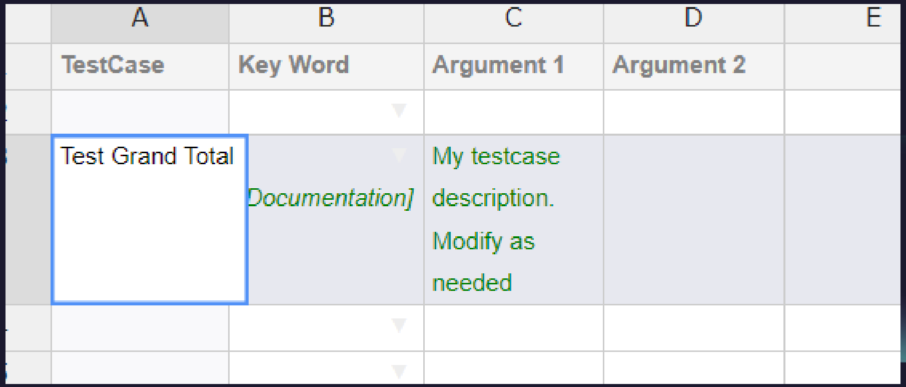
	A	B	C	D	E	F	G	H	I
1	TestCase	Key Word	Argument 1	Argument 2					
2									
3	My First Testcase	[Documentation]	My testcase description. Modify as needed						
4		My Function	my argument	my argument 2					
5									
6									
7									



Script Directory: C:/SahiPro/userdata/scripts/

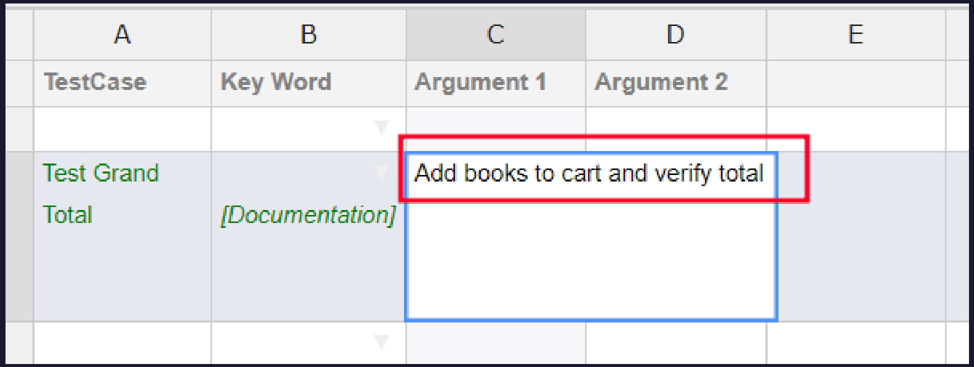
File: cart

- Give a name in the test case column.



	A	B	C	D	E
	TestCase	Key Word	Argument 1	Argument 2	
	Test Grand Total	[Documentation]	My testcase description. Modify as needed		

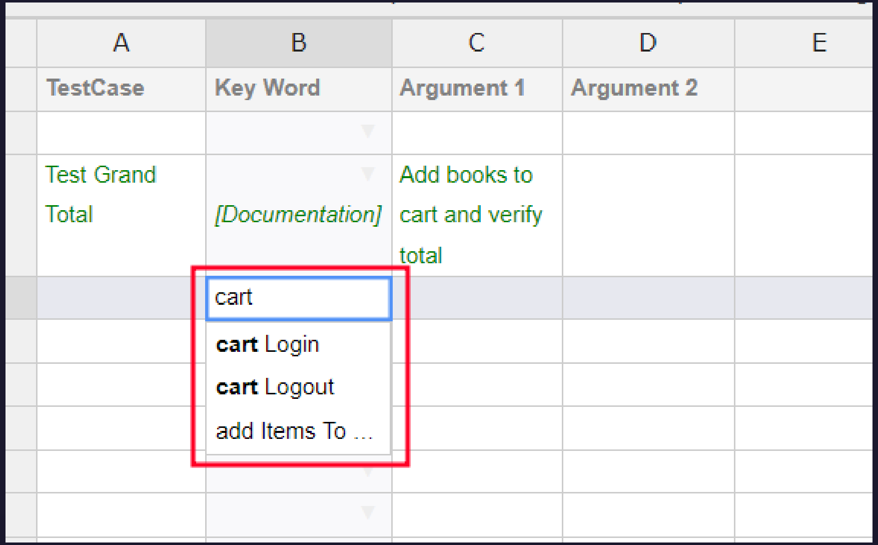
- Add a description to the test case in Argument 1 column.



	A	B	C	D	E
	TestCase	Key Word	Argument 1	Argument 2	
	Test Grand Total	[Documentation]	Add books to cart and verify total		

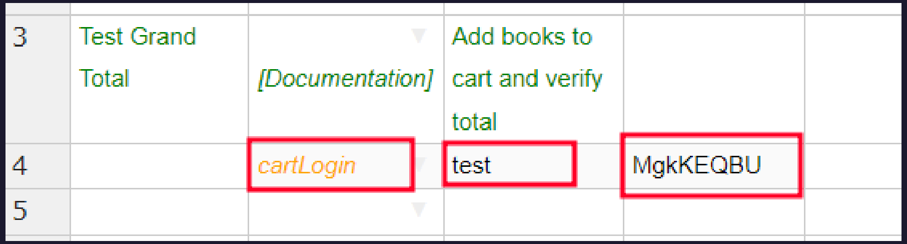
- User can use different functions available in cart_library file.

- When user starts to type the function name, Sahi shows the available functions.



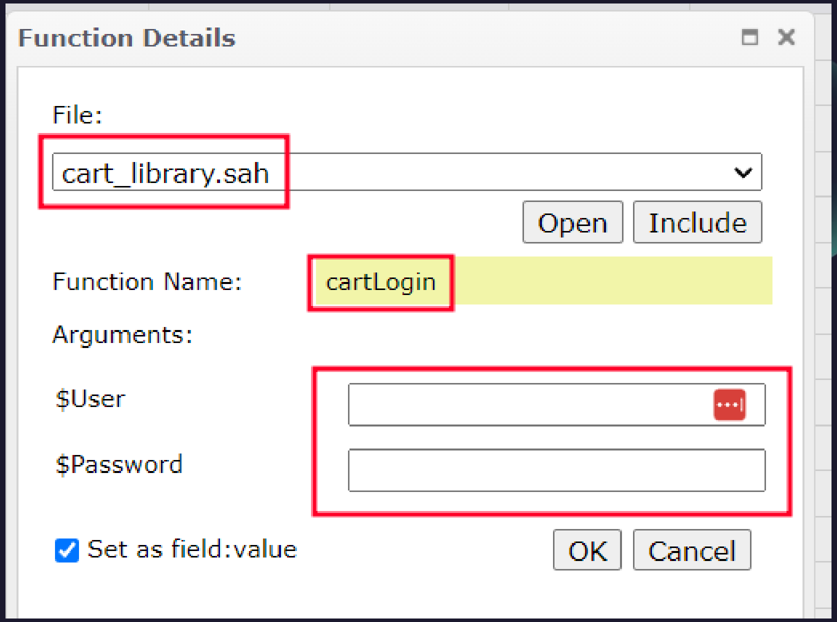
	A	B	C	D	E
	TestCase	Key Word	Argument 1	Argument 2	
	Test Grand Total	[Documentation]	Add books to cart and verify total		
		cart			

- Ctrl+click on the function to know more details about it
- Select Include button to include cart_library.sah in the current scenario file.
- Now user can enter the data that is needed for login to happen i.e., username and password in Argument 1 and Argument 2 columns



3	Test Grand Total	[Documentation]	Add books to cart and verify total	
4		cartLogin	test	MgkKEQBU
5				

- User can pass the parameters values in Function Details window.



Function Details

File: cart_library.sah

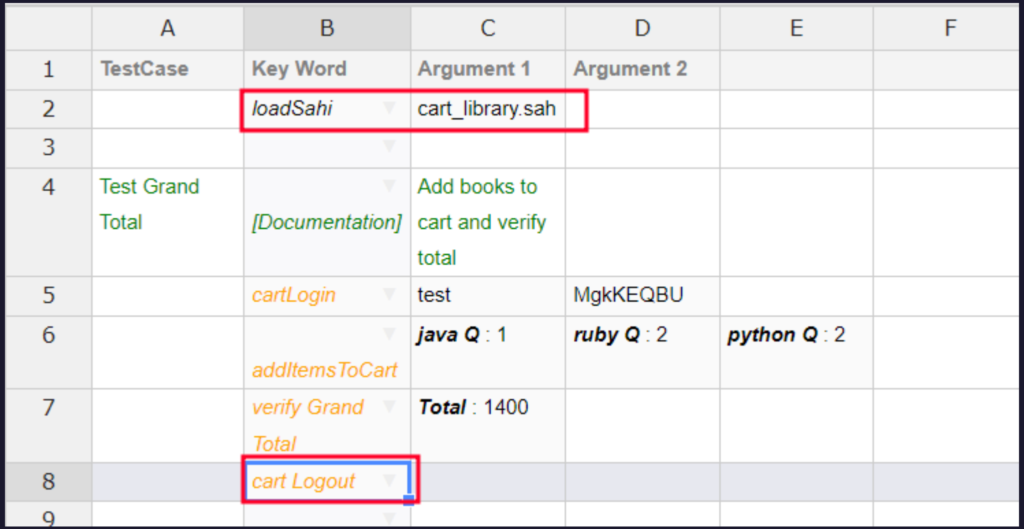
Function Name: cartLogin

Arguments:

$User

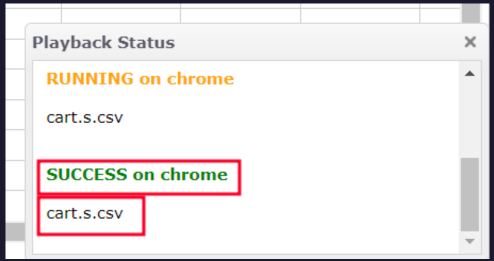
$Password

[✓] Set as field:value



	A	B	C	D	E	F
1	TestCase	Key Word	Argument 1	Argument 2		
2		loadSahi	cart_library.sah			
3						
4	Test Grand Total	[Documentation]	Add books to cart and verify total			
5		cartLogin	test	MgkKEQBU		
6		addItemsToCart	***java Q*** : 1	***ruby Q*** : 2	***python Q*** : 2	
7		verify Grand Total	***Total*** : 1400			
8		cart Logout				
9						

- Let's run the script

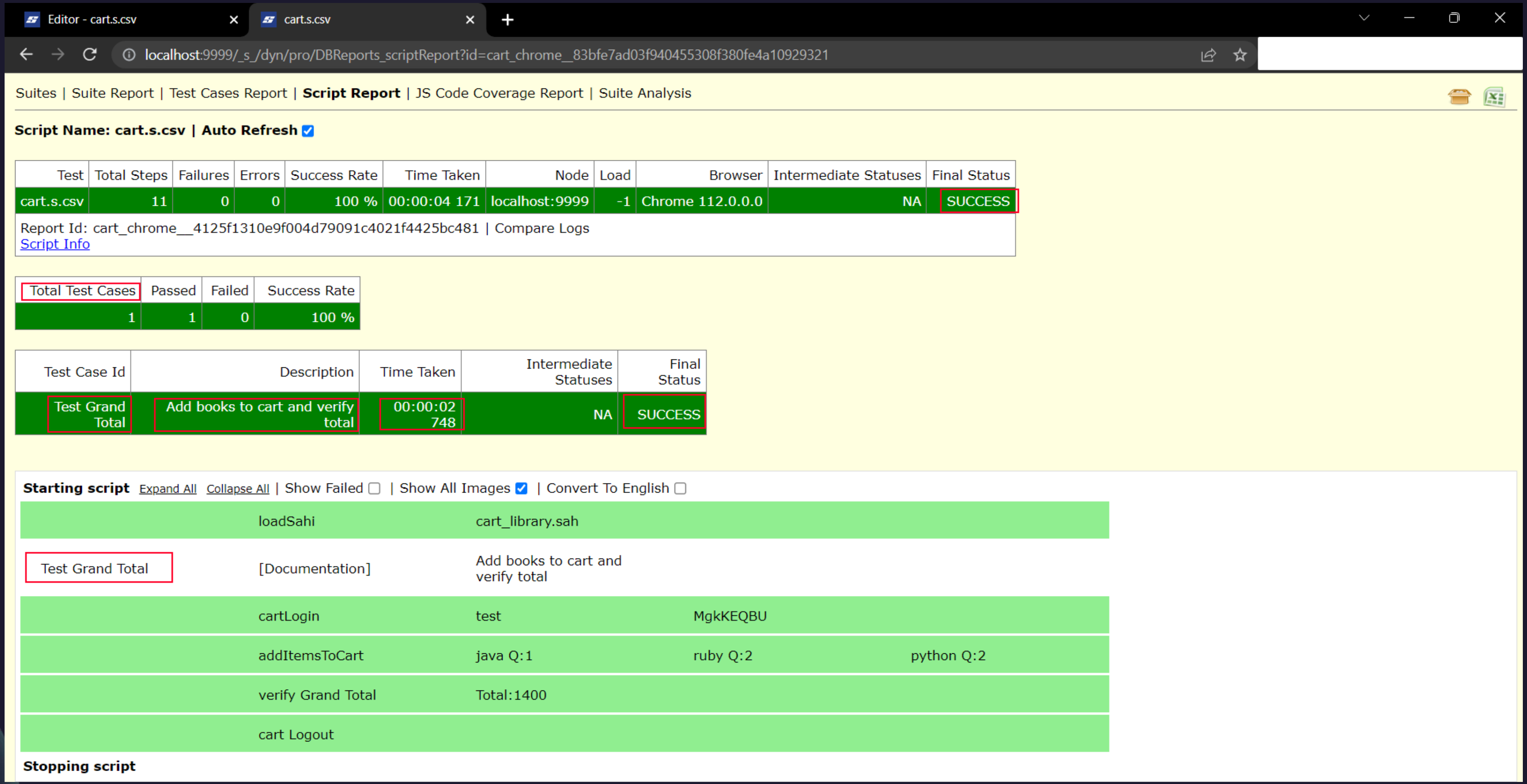

Playback Status

RUNNING on chrome

cart.s.csv

SUCCESS on chrome

cart.s.csv


Editor - cart.s.csv
cart.s.csv
localhost:9999/_s_/dyn/pro/DBReports_scriptReport?id=cart_chrome__83bfe7ad03f940455308f380fe4a10929321
Suites | Suite Report | Test Cases Report | Script Report | JS Code Coverage Report | Suite Analysis

Script Name: cart.s.csv | Auto Refresh ✓

Test	Total Steps	Failures	Errors	Success Rate	Time Taken	Node	Load	Browser	Intermediate Statuses	Final Status
cart.s.csv	11	0	0	100 %	00:00:04 171	localhost:9999	-1	Chrome 112.0.0.0	NA	SUCCESS

Report Id: cart_chrome__4125f1310e9f004d79091c4021f4425bc481 | Compare Logs
Script Info

Total Test Cases	Passed	Failed	Success Rate
1	1	0	100 %

Test Case Id	Description	Time Taken	Intermediate Statuses	Final Status
Test Grand Total	Add books to cart and verify total	00:00:02 748	NA	SUCCESS

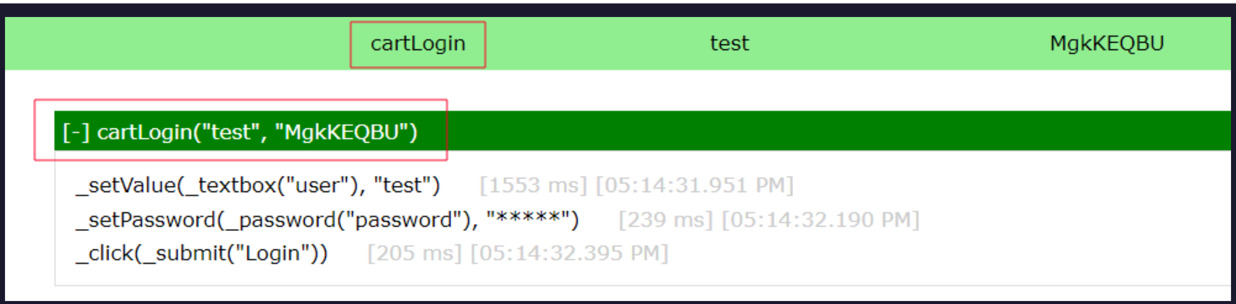
Starting script Expand All Collapse All | Show Failed ☐ | Show All Images ✓ | Convert To English ☐

	loadSahi	cart_library.sah		
Test Grand Total	[Documentation]	Add books to cart and verify total		
	cartLogin	test	MgkKEQBU	
	addItemsToCart	java Q:1	ruby Q:2	python Q:2
	verify Grand Total	Total:1400		
	cart Logout			

Stopping script

- Opening logs

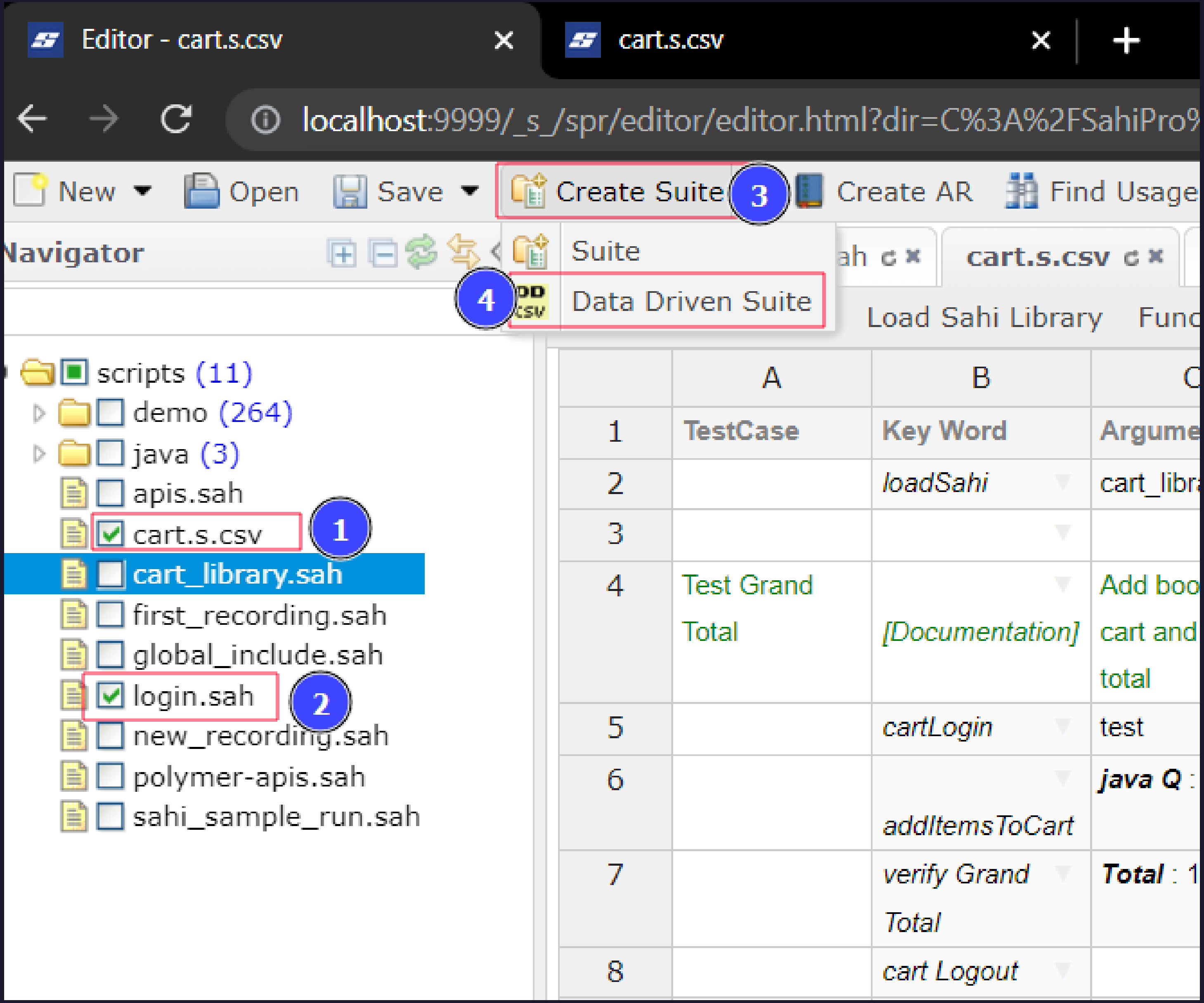
- Clicking the step name shows the function called and the code executed.


cartLogin test MgkKEQBU

```
[-] cartLogin("test", "MgkKEQBU")
_setValue(_textbox("user"), "test")   [1553 ms] [05:14:31.951 PM]
_setPassword(_password("password"), "*****")   [239 ms] [05:14:32.190 PM]
_click(_submit("Login"))   [205 ms] [05:14:32.395 PM]
```

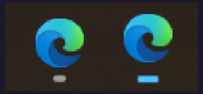
Data Driven Suites

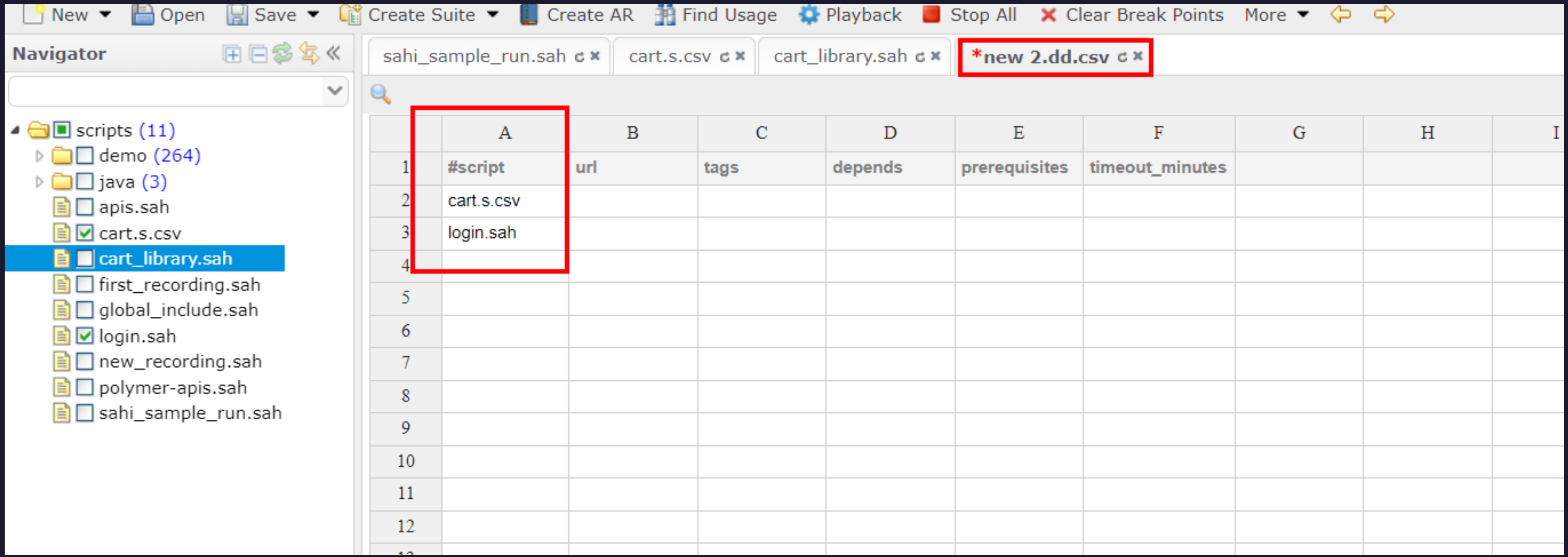
- Let's run our scripts in batches using Sahi Parallel playback.
- To execute multiple scripts in a batch, we need to create a suite file.
- We create a suite file by first choosing the scripts that we want to execute.
- Now click on Create Suite → Data Driven Suite



	A	B	C
1	TestCase	Key Word	Argume
2		loadSahi	cart_libra
3			
4	Test Grand Total	[Documentation]	Add boo cart and total
5		cartLogin	test
6		addItemsToCart	java Q :
7		verify Grand Total	Total : 1
8		cart Logout	

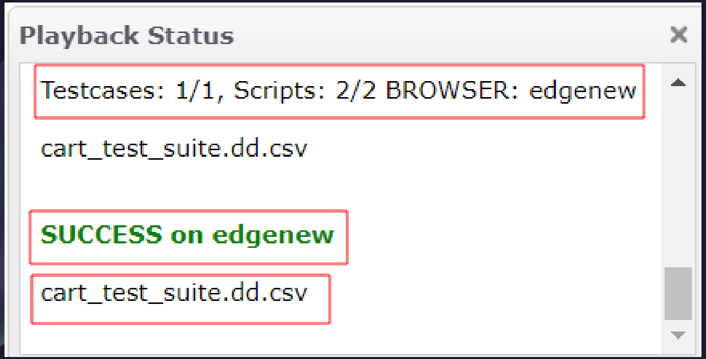
- The above step creates a suite file. Save the file.
- Let's playback the suite using 'Playback' button.
- Playback happens parallelly opening 2 edge browsers.





scripts (11)
demo (264)
java (3)
apis.sah
cart.s.csv
cart_library.sah
first_recording.sah
global_include.sah
login.sah
new_recording.sah
polymer-apis.sah
sahi_sample_run.sah

	A	B	C	D	E	F	G	H
1	#script	url	tags	depends	prerequisites	timeout_minutes		
2	cart.s.csv							
3	login.sah							



Testcases: 1/1, Scripts: 2/2 BROWSER: edgenew

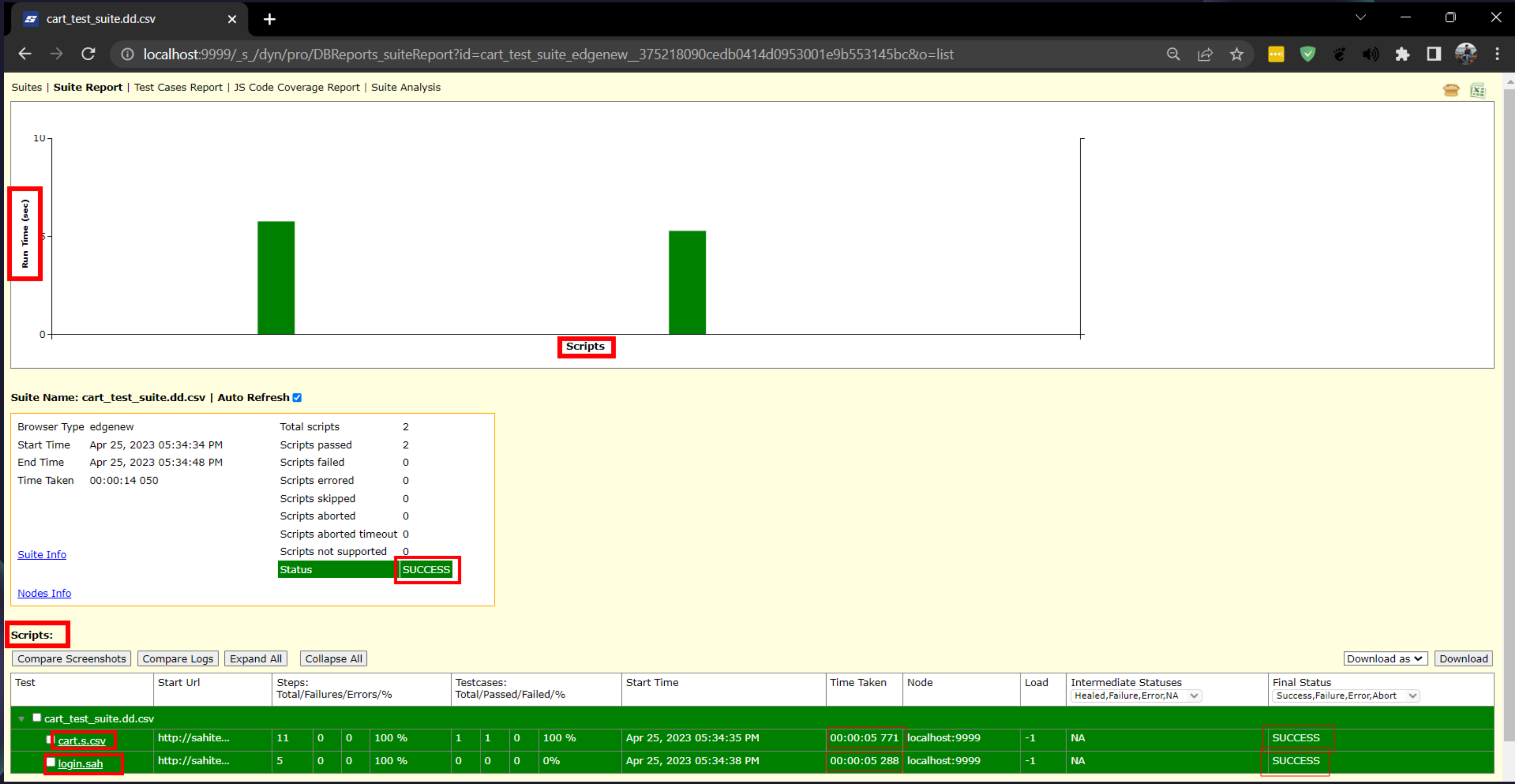
cart_test_suite.dd.csv

SUCCESS on edgenew

cart_test_suite.dd.csv

Data Driven Suites Logs

- Logs show the graph showing the scripts and their run time.
- User can see the time taken for running the scripts.
- Clicking on the test link, Sahi opens detailed report of the execution.


cart_test_suite.dd.csv
localhost:9999/_s_/dyn/pro/DBReports_suiteReport?id=cart_test_suite_edgenew__375218090cedb0414d0953001e9b553145bc&o=list
Suites | Suite Report | Test Cases Report | JS Code Coverage Report | Suite Analysis
10
5
0
Run Time (sec)
Scripts

Suite Name: cart_test_suite.dd.csv | Auto Refresh ☑

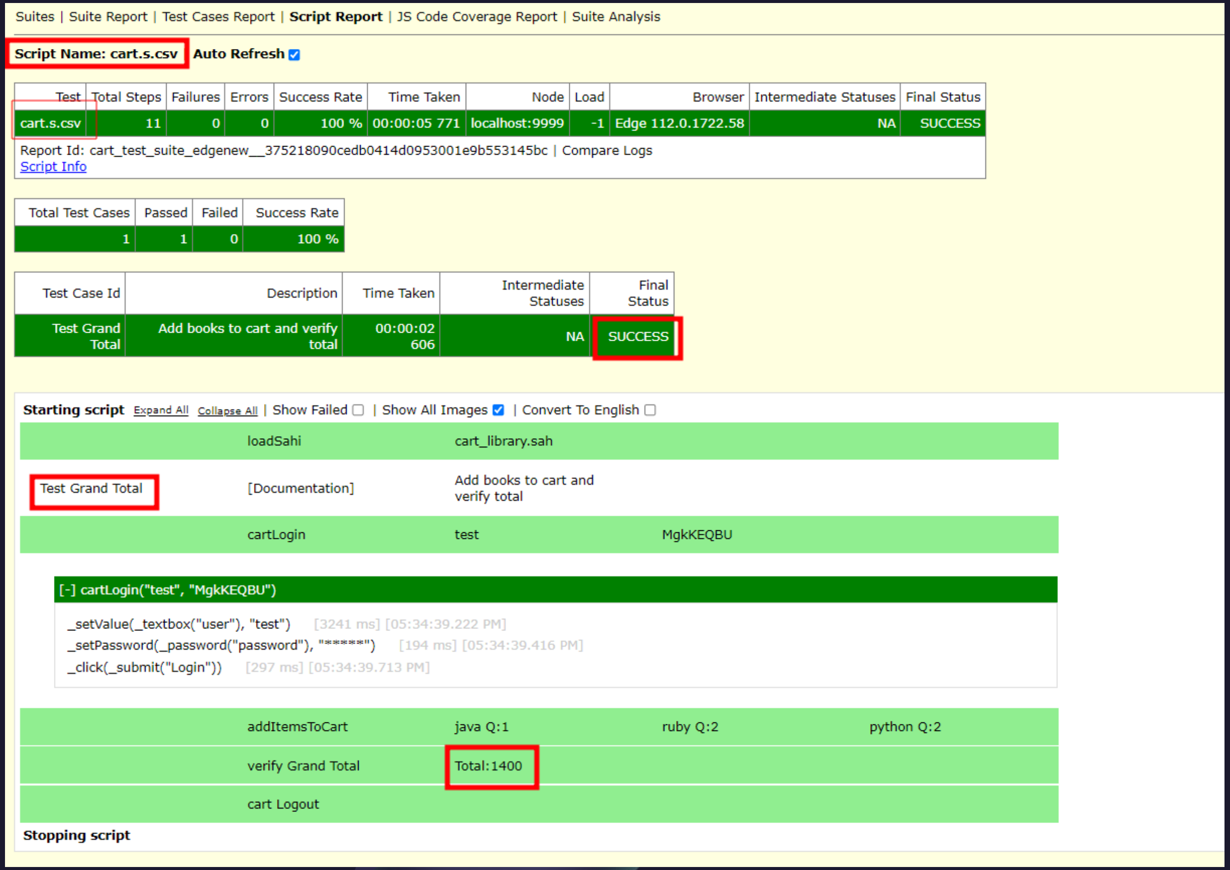
Browser Type	edgenew	Total scripts	2
Start Time	Apr 25, 2023 05:34:34 PM	Scripts passed	2
End Time	Apr 25, 2023 05:34:48 PM	Scripts failed	0
Time Taken	00:00:14 050	Scripts errored	0
		Scripts skipped	0
		Scripts aborted	0
		Scripts aborted timeout	0
Suite Info		Scripts not supported	0
		Status	SUCCESS
Nodes Info			

Scripts:

Compare Screenshots
Compare Logs
Expand All
Collapse All
Download as
Download

Test	Start Url	Steps: Total/Failures/Errors/%				Testcases: Total/Passed/Failed/%				Start Time	Time Taken	Node	Load	Intermediate Statuses Healed,Failure,Error,NA	Final Status Success,Failure,Error,Abort
cart_test_suite.dd.csv															
cart.s.csv	http://sahite...	11	0	0	100 %	1	1	0	100 %	Apr 25, 2023 05:34:35 PM	00:00:05 771	localhost:9999	-1	NA	SUCCESS
login.sah	http://sahite...	5	0	0	100 %	0	0	0	0%	Apr 25, 2023 05:34:38 PM	00:00:05 288	localhost:9999	-1	NA	SUCCESS

Data Driven Suites Logs Detailed View



Suites | Suite Report | Test Cases Report | **Script Report** | JS Code Coverage Report | Suite Analysis

Script Name: cart.s.csv **Auto Refresh** ☑

Test	Total Steps	Failures	Errors	Success Rate	Time Taken	Node	Load	Browser	Intermediate Statuses	Final Status
cart.s.csv	11	0	0	100 %	00:00:05 771	localhost:9999	-1	Edge 112.0.1722.58	NA	SUCCESS

Report Id: cart_test_suite_edgenew__375218090cedb0414d0953001e9b553145bc | Compare Logs
Script Info

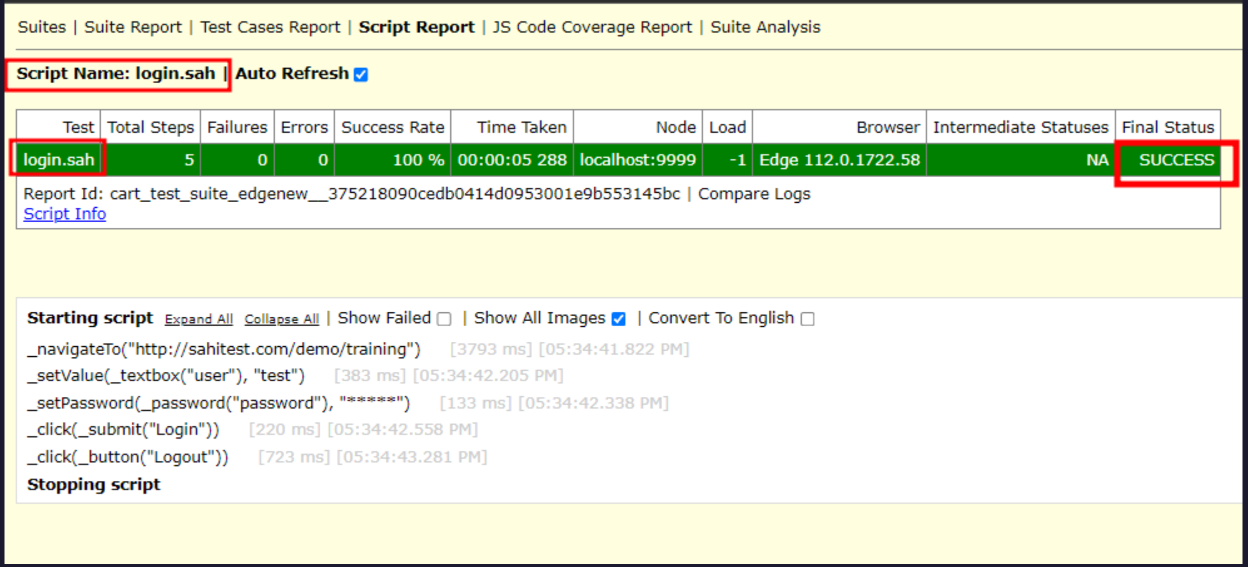
Total Test Cases	Passed	Failed	Success Rate
1	1	0	100 %

Test Case Id	Description	Time Taken	Intermediate Statuses	Final Status
Test Grand Total	Add books to cart and verify total	00:00:02 606	NA	SUCCESS

Starting script Expand All Collapse All | Show Failed ☐ | Show All Images ☑ | Convert To English ☐

```
                    loadSahi              cart_library.sah
Test Grand Total    [Documentation]       Add books to cart and
                                          verify total
                    cartLogin             test                MgkKEQBU

    [-] cartLogin("test", "MgkKEQBU")
    _setValue(_textbox("user"), "test")    [3241 ms] [05:34:39.222 PM]
    _setPassword(_password("password"), "******")    [194 ms] [05:34:39.416 PM]
    _click(_submit("Login"))    [297 ms] [05:34:39.713 PM]

                    addItemsToCart        java Q:1            ruby Q:2            python Q:2
                    verify Grand Total    Total:1400
                    cart Logout
```

Stopping script



Suites | Suite Report | Test Cases Report | **Script Report** | JS Code Coverage Report | Suite Analysis

Script Name: login.sah | **Auto Refresh** ☑

Test	Total Steps	Failures	Errors	Success Rate	Time Taken	Node	Load	Browser	Intermediate Statuses	Final Status
login.sah	5	0	0	100 %	00:00:05 288	localhost:9999	-1	Edge 112.0.1722.58	NA	SUCCESS

Report Id: cart_test_suite_edgenew__375218090cedb0414d0953001e9b553145bc | Compare Logs
Script Info

Starting script Expand All Collapse All | Show Failed ☐ | Show All Images ☑ | Convert To English ☐

```
_navigateTo("http://sahitest.com/demo/training")    [3793 ms] [05:34:41.822 PM]
_setValue(_textbox("user"), "test")    [383 ms] [05:34:42.205 PM]
_setPassword(_password("password"), "******")    [133 ms] [05:34:42.338 PM]
_click(_submit("Login"))    [220 ms] [05:34:42.558 PM]
_click(_button("Logout"))    [723 ms] [05:34:43.281 PM]
```

Stopping script

Business-Driven Test Automation

- Business-Driven Test Automation (BDTA) allows test automation to begin much earlier in the project lifecycle - right at the feature conceptualization stage. The application under test need not be ready to begin BDTA.
- With BDTA, prior to the readiness of your feature or application, we can:
  - Specify the desired application flow using straightforward language, employing key words for various actions (e.g., login, add books, etc.).
  - Incorporate optional parameters into different steps based on specific testing requirements, allowing for future adjustments or additions when the application is fully developed (e.g., login | username: test | password: |).

Thank You

Naga Vara Pradeep Yendluri

Group 11