AutonomIQ

Release 5.1

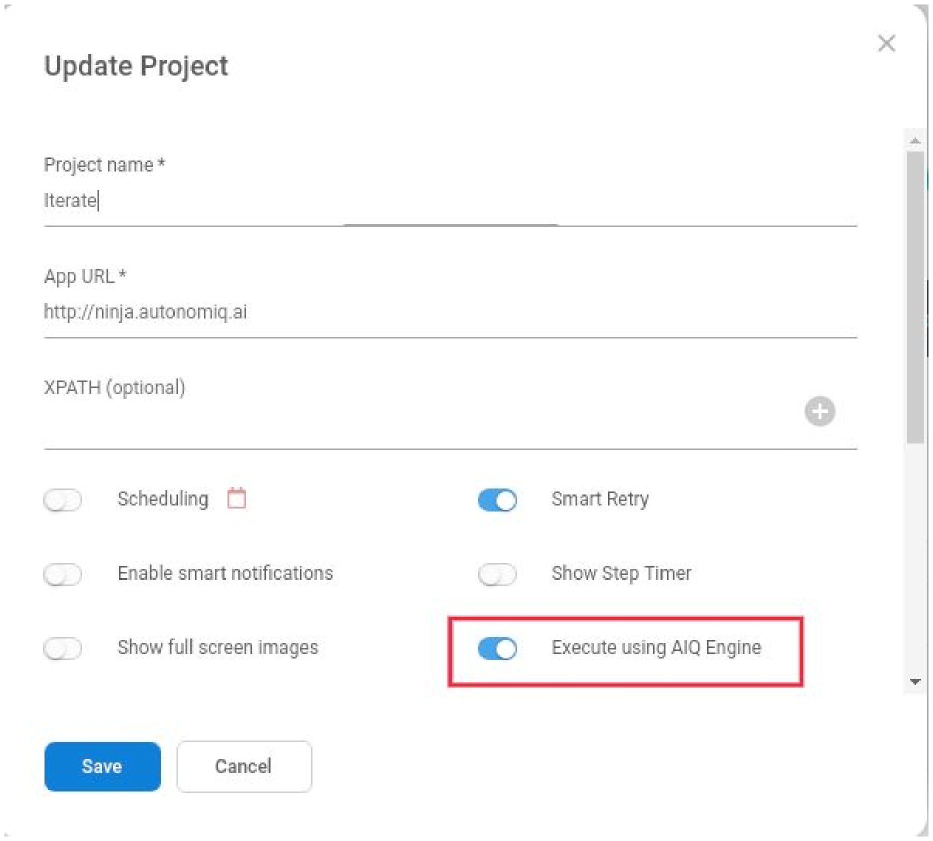
Data Driven for AIQ

In Data-driven test, input data can be stored in data sources like xls, csv and the test case which can execute tests for all test data in the xls, csv. I.e - run through multiple data for input in a for-each loop

Earlier this feature was supported only by selenium mode now it is also supported by AIQ mode execution

Note: While generating, only the first loop is used. During execution, all the data will be used and looped through.

AutonomIQ


Update Project

Project name * Iterate

App URL * http://ninja.autonomiq.ai

XPATH (optional)

Scheduling: off

Smart Retry: on

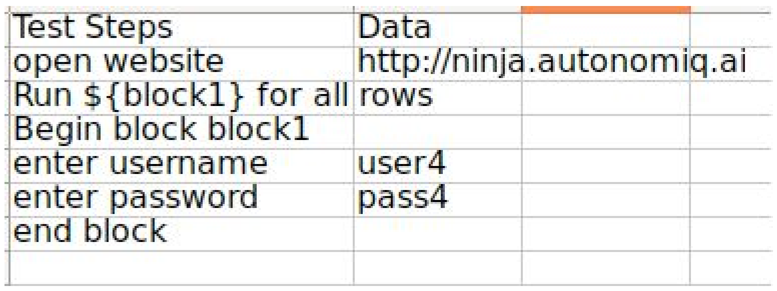
Enable smart notifications: off

Show Step Timer: off

Show full screen images: off

Execute using AIQ Engine: on

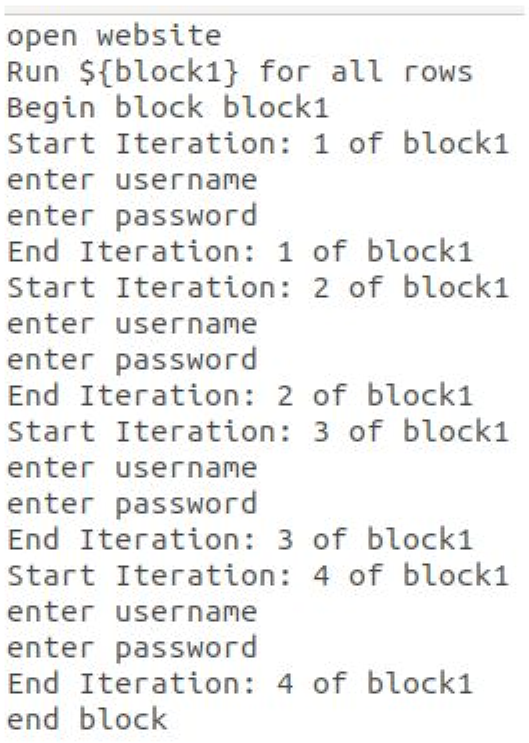
Data Driven Sample Test Case



Test Steps	Data		
open website	http://ninja.autonomiq.ai		
Run ${block1} for all	rows		
Begin block block1			
enter username	user4		
enter password	pass4		
end block			

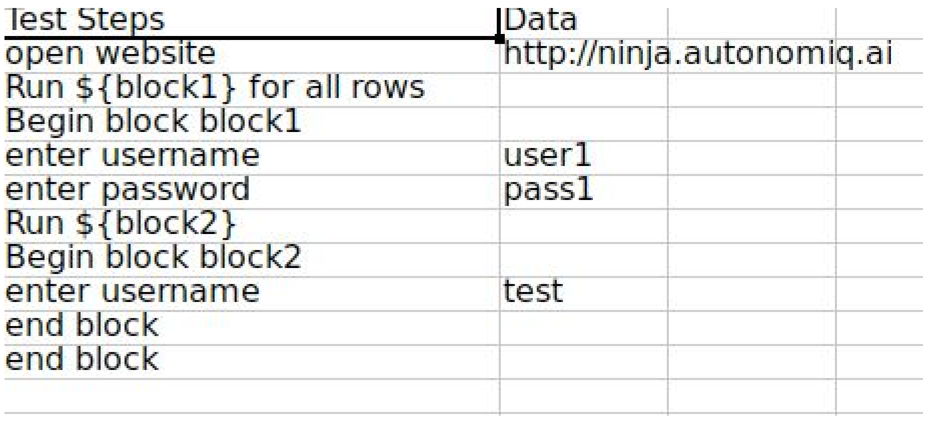
AutonomIQ

Report which gives clear understanding of start and end iteration for each block


```
open website
Run ${block1} for all rows
Begin block block1
Start Iteration: 1 of block1
enter username
enter password
End Iteration: 1 of block1
Start Iteration: 2 of block1
enter username
enter password
End Iteration: 2 of block1
Start Iteration: 3 of block1
enter username
enter password
End Iteration: 3 of block1
Start Iteration: 4 of block1
enter username
enter password
End Iteration: 4 of block1
end block
```

Nested Blocks

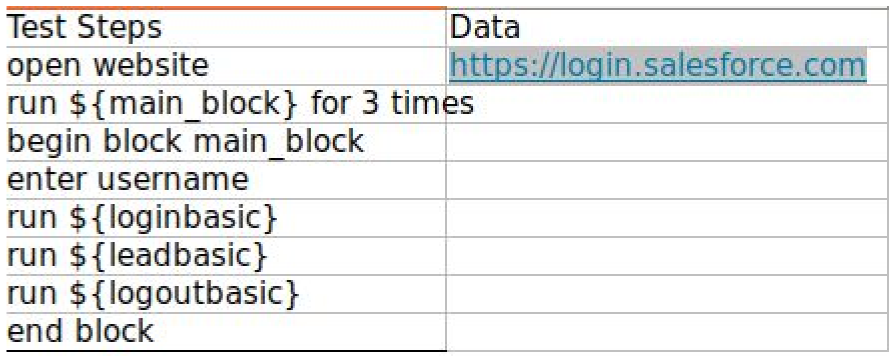
Nested blocks are blocks within blocks. You can have a single level of nesting, or you can even have multiple levels of nesting blocks



Test Steps	Data
open website	http://ninja.autonomiq.ai
Run ${block1} for all rows	
Begin block block1	
enter username	user1
enter password	pass1
Run ${block2}	
Begin block block2	
enter username	test
end block	
end block	

Flows inside a block is also supported

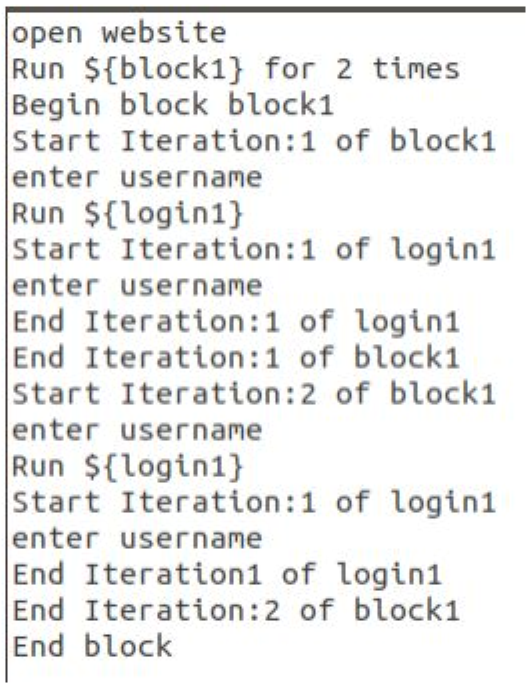
Here's a sample test case for nested flows



Test Steps	Data
open website	https://login.salesforce.com
run ${main_block} for 3 times	
begin block main_block	
enter username	
run ${loginbasic}	
run ${leadbasic}	
run ${logoutbasic}	
end block	

The corresponding flows - loginbasic, leadbasic and logoutbasic have to be created under the flows tab as per user guide.

Sample report format for nested flows



```
open website
Run ${block1} for 2 times
Begin block block1
Start Iteration:1 of block1
enter username
Run ${login1}
Start Iteration:1 of login1
enter username
End Iteration:1 of login1
End Iteration:1 of block1
Start Iteration:2 of block1
enter username
Run ${login1}
Start Iteration:1 of login1
enter username
End Iteration1 of login1
End Iteration:2 of block1
End block
```

Main Block used to repeat a given section of block/flow a certain number of times or until a particular condition is met.Iteration count of Main Block is shown in the increasing order in the report eg: Start Iteration1 of block1, Start Iteration2 of block1 where its respective sub block/flow will always start from count 1

Note: We request user to create test case in below order

```
Run ${block1}
Begin block block1
instructions
End block
```

Decision Making Statement for Blocks/Flows

Note: Decision making statement (i.e) if and the else part will work only for block and flow statement

If statement

if statement is the most simple decision making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statements is executed otherwise not.

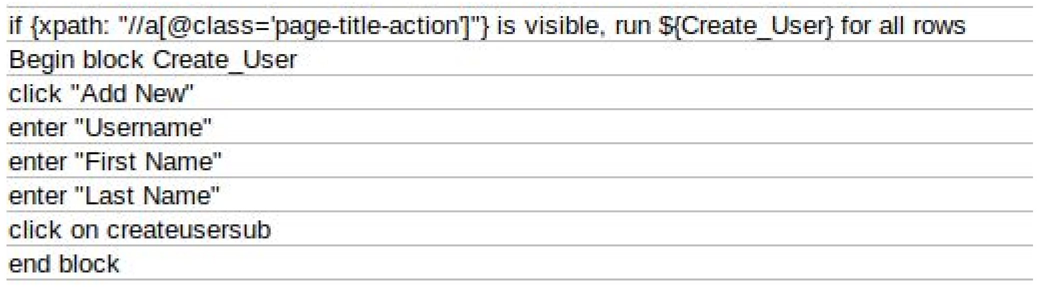
Syntax

```
if(condition), run ${block}
Begin block blockname
   // Statements to execute if
   // condition is true
End block
```

The condition can be used with flow as below, we can call the flow or can create a block

```
if(condition), run ${flow}
   // Statements to execute if
   // condition is true
```

Example:



if {xpath: "//a[@class='page-title-action']"} is visible, run ${Create_User} for all rows
Begin block Create_User
click "Add New"
enter "Username"
enter "First Name"
enter "Last Name"
click on createusersub
end block

If the given xpath is visible then users will be created.

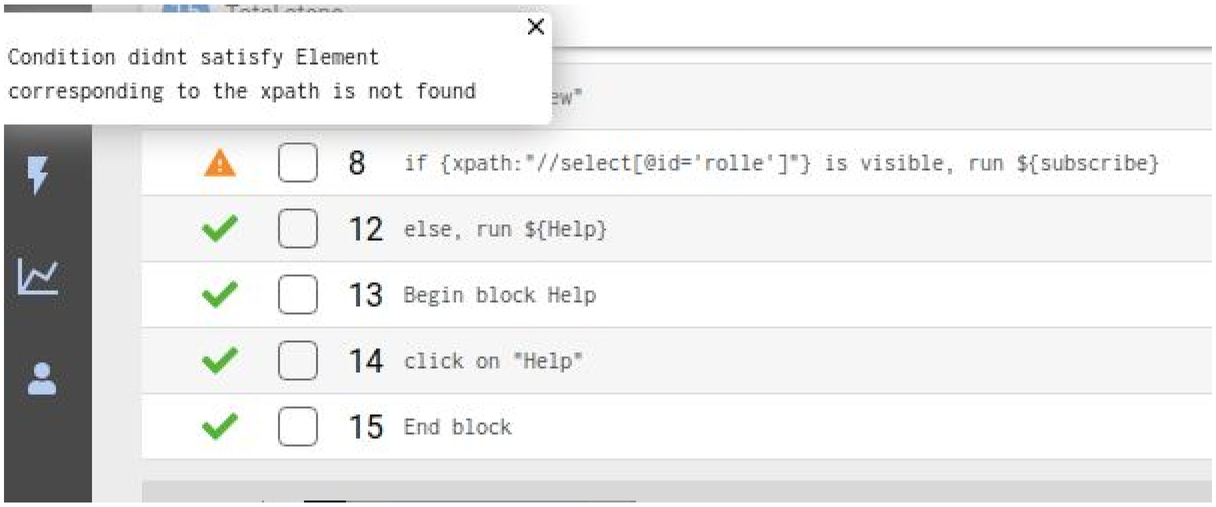
If-else statements

If-else statement, if a condition is true a block of statements will be executed and if the condition is false else part will be executed

Syntax

```
if (condition) run ${block}
Begin block block name
    // Executes this block if
    // condition is true
End Block
Else, run ${else_part}
Begin block else_part
    // Executes this block if
    // condition is false
End Block
```

Example:



Condition didnt satisfy Element corresponding to the xpath is not found

```
8   if {xpath:"//select[@id='rolle']"} is visible, run ${subscribe}
12  else, run ${Help}
13  Begin block Help
14  click on "Help"
15  End block
```

In this example the if condition is not satisfied so else part is executed

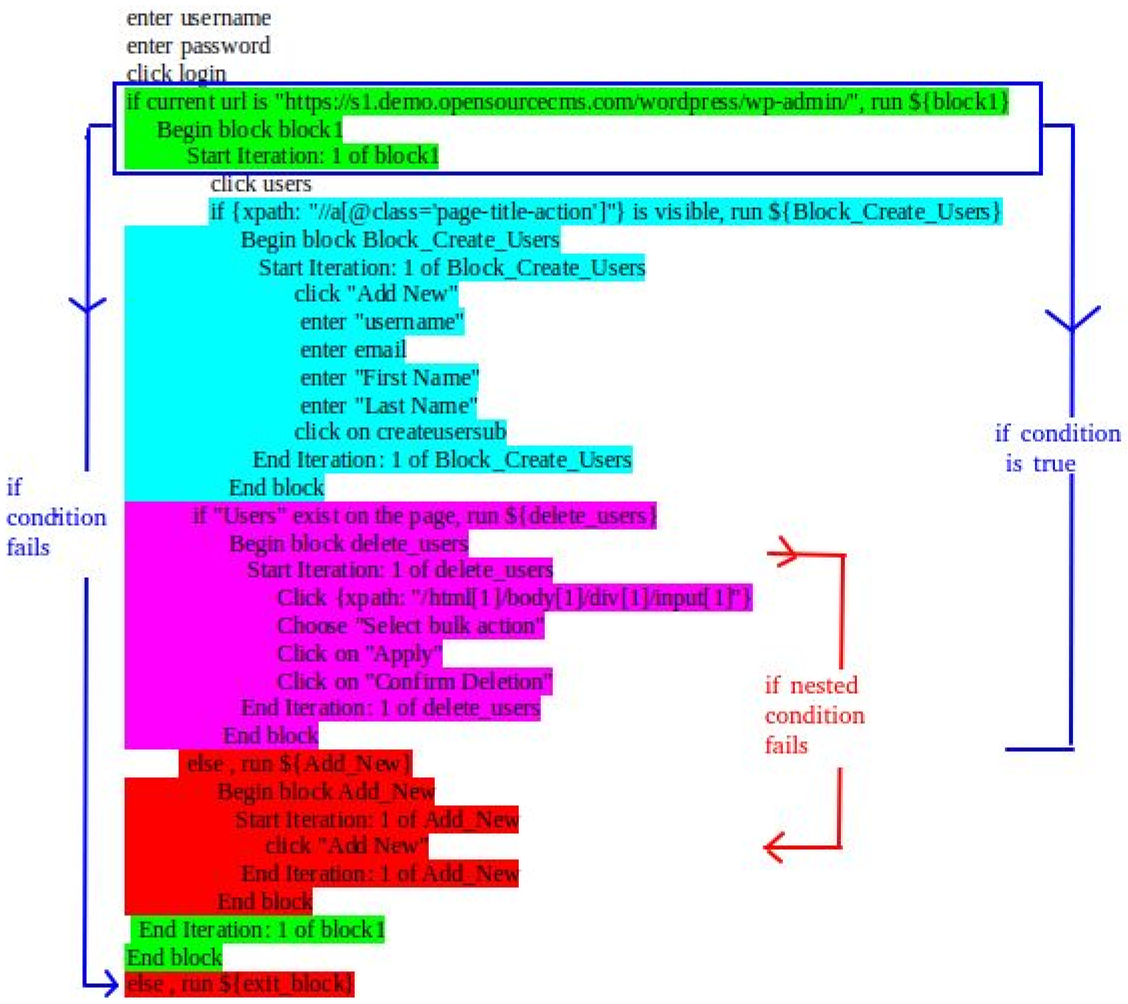
Nested if statements

When an if else statement is present inside the body of another “if” or “else” then this is called nested if else.

Syntax:

```
if (condition1), run ${block1}
Begin block block1
    //Nested if else inside the body of "if"
    if(condition2), run ${block2}
    Begin block block2
       //Statements inside the body of nested "if"
    End Block
    Else, ${else_part}
       // else_part is the flow here
       //Statements inside the body of nested "else"
Else, ${else_mainblock}
    //Statements inside the body of "else"
```

Example:



```
enter username
enter password
click login
if current url is "https://s1.demo.opensourcecms.com/wordpress/wp-admin/", run ${block1}
    Begin block block1
        Start Iteration: 1 of block1
            click users
            if {xpath: "//a[@class='page-title-action']"} is visible, run ${Block_Create_Users}
                Begin block Block_Create_Users
                  Start Iteration: 1 of Block_Create_Users
                      click "Add New"
                       enter "username"
                       enter email
                       enter "First Name"
                       enter "Last Name"
                      click on createusersub
                 End Iteration: 1 of Block_Create_Users
              End block
         if "Users" exist on the page, run ${delete_users}
              Begin block delete_users
                Start Iteration: 1 of delete_users
                    Click {xpath: "/html[1]/body[1]/div[1]/input[1]"}
                    Choose "Select bulk action"
                    Click on "Apply"
                    Click on "Confirm Deletion"
                End Iteration: 1 of delete_users
             End block
        else , run ${Add_New}
            Begin block Add_New
              Start Iteration: 1 of Add_New
                  click "Add New"
               End Iteration: 1 of Add_New
            End block
 End Iteration: 1 of block1
End block
else , run ${exit_block}
```

AutonomIQ

Else-if (elif)

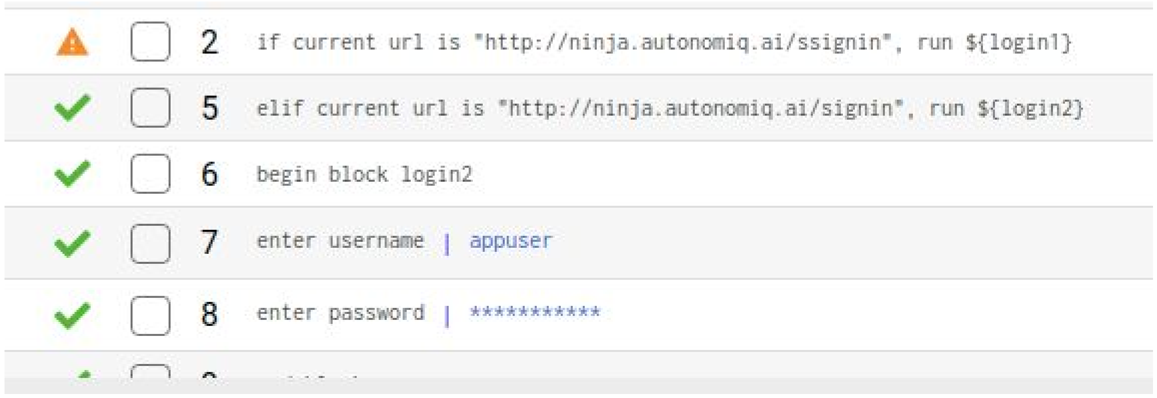
The elif statement is useful when you need to check multiple conditions, nesting of if-else blocks can be avoided using else..if statement.

Note: instead of **else if** we need to mentioned as **elif**

Syntax:

```
if (condition1) run ${block1}
   //These statements would execute if the condition1 is true
elif(condition2) run ${block2}
   //These statements would execute if the condition2 is true and condition1 is false
End block
.
.
Else, run ${else_block}
   //These statements would execute if all the conditions return false.
End block
End block
```

Example:



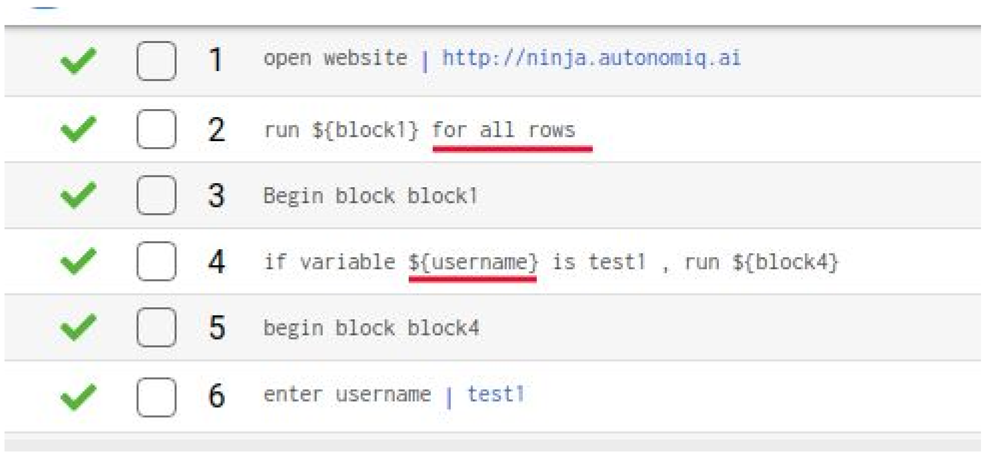
```
2  if current url is "http://ninja.autonomiq.ai/ssignin", run ${login1}
5  elif current url is "http://ninja.autonomiq.ai/signin", run ${login2}
6  begin block login2
7  enter username | appuser
8  enter password | ***********
```

In this example if condition is not satisfied so the block **login1** is not executed, then the control moves to elif here the condition is satisfied and the block **login2** is executed , when at least one condition is passed the else part will be skipped

AutonomIQ

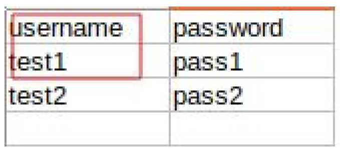
Data Driven Parsing when condition satisfied

Data will be parsed when condition is satisfied



✓	1	open website | http://ninja.autonomiq.ai
✓	2	run ${block1} for all rows
✓	3	Begin block block1
✓	4	if variable ${username} is test1 , run ${block4}
✓	5	begin block block4
✓	6	enter username | test1

DataFile



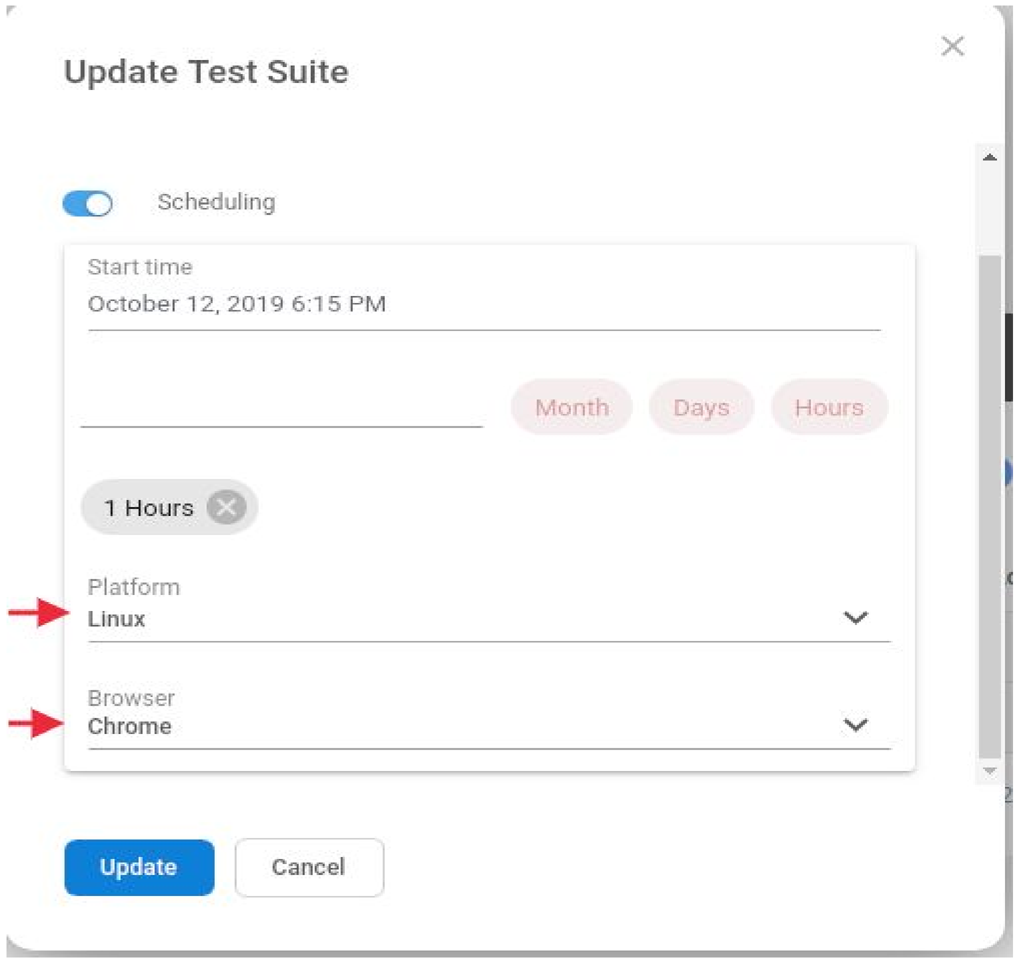
username	password
test1	pass1
test2	pass2

Above testcase run${block1} for all rows will iterate through all rows in the data file. When condition matches for the current row that is running now, only that if block will get executed, Subsequent elif/else wont get executed, likewise if the condition did not match for other rows that if wont get executed.

AutonomIQ

Schedule in Different Browsers

While scheduling a suite user have option to select platform and browser details so that the scheduled suite will execute in the respective platform, browser.



Update Test Suite

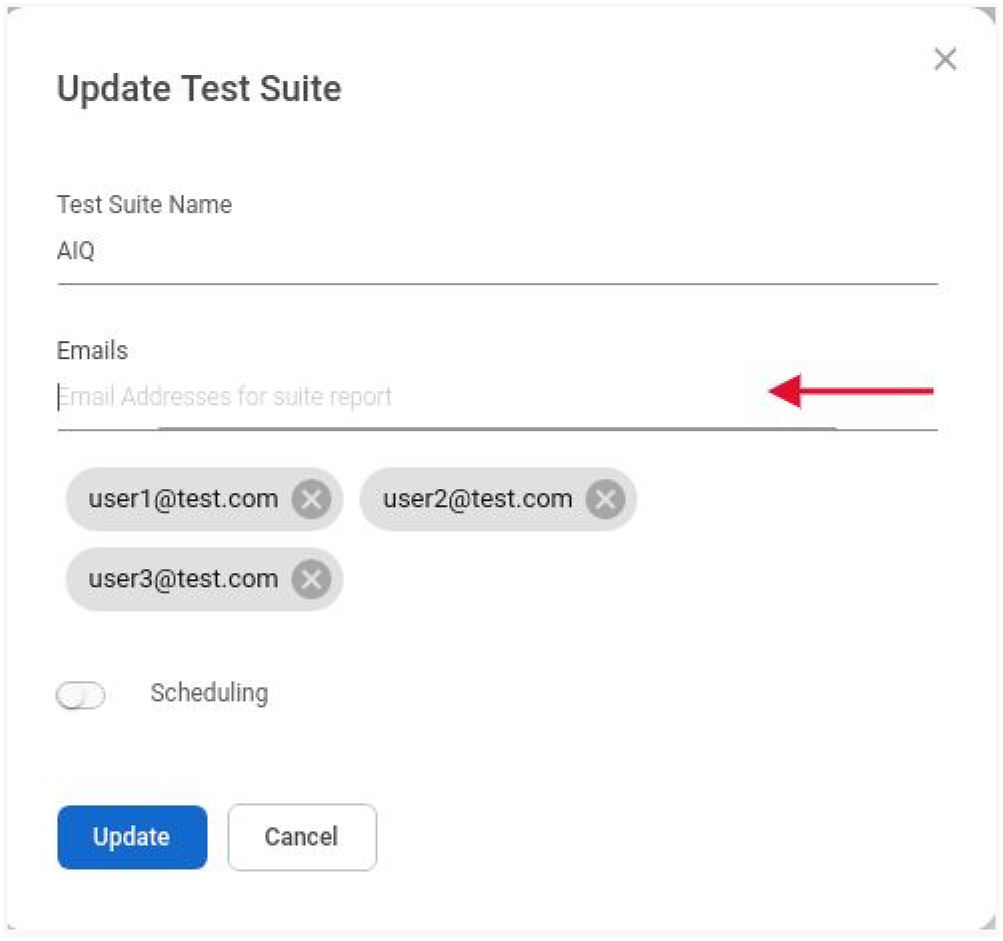
Scheduling

Start time
October 12, 2019 6:15 PM

Platform
Linux

Browser
Chrome

Ability to add multiple emails to a suite for receiving consolidated reports


Update Test Suite

Test Suite Name
AIQ

Emails
Email Addresses for suite report

user1@test.com
user2@test.com
user3@test.com

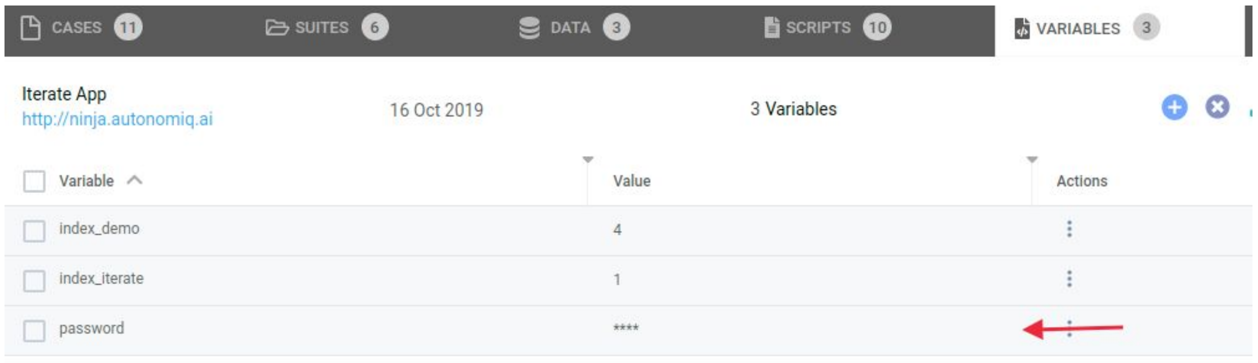
User can now add multiple email to a suite for receiving consolidated report after the suite execution. Enter valid email and by press tab,comma or enter key multiple emails can be added.

Hide Password in Variable and Data

Password given in the variable and in data will be hidden. That is sensitive data are hidden, only by downloading the variable/data the password details can be seen.

AutonomIQ

Note: we now hide only value for statement that have word “password”. In the future we will apply this for other common sensitive words "pass", "pwd", "user", "userid", "login", "username", "uid" that are in common.



Iterate App
http://ninja.autonomiq.ai

16 Oct 2019

3 Variables

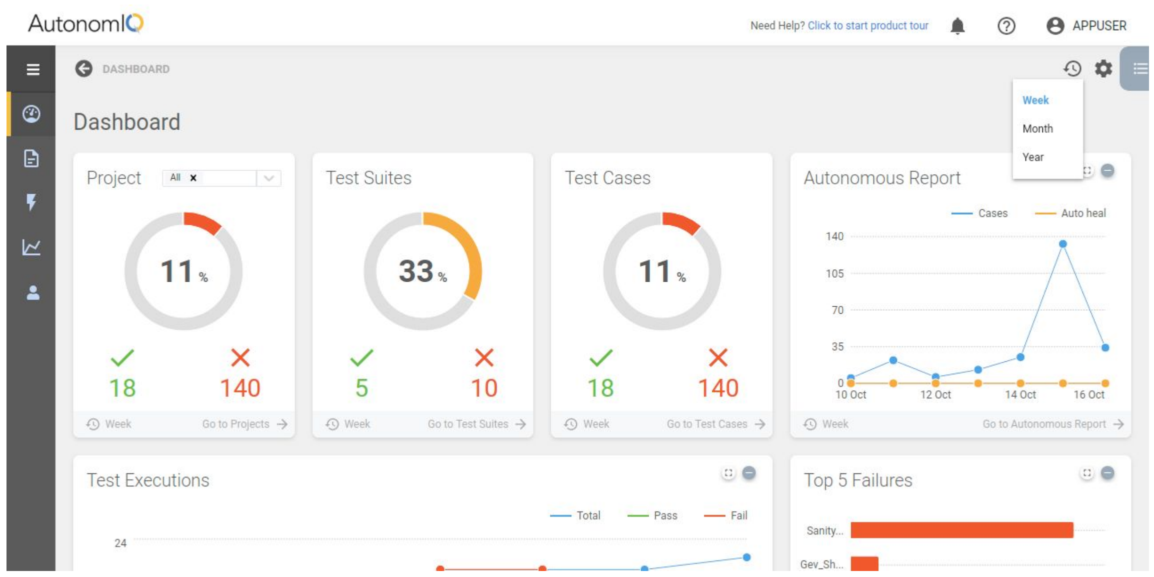
Variable	Value	Actions
index_demo	4	⋮
index_iterate	1	⋮
password	****	⋮

AutonomIQ

UI Changes

New Dashboard

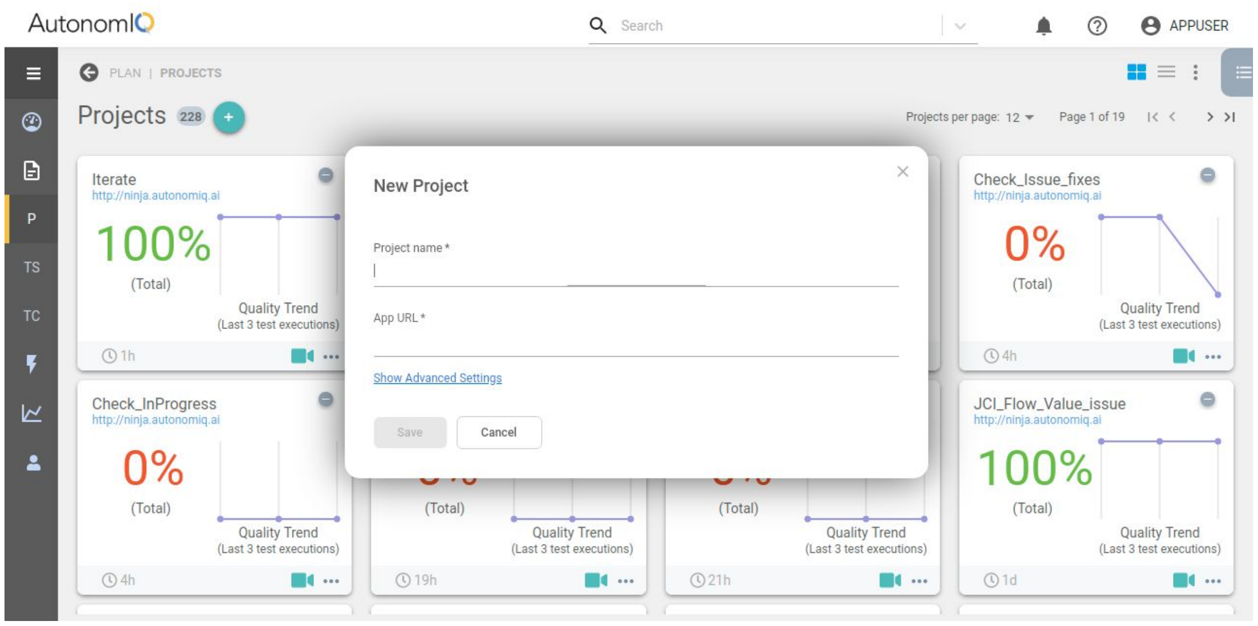
Dashboard allows us to check Statistical data for Week, Month and Year with the new UI graphical user interface.



AutonomIQ

New Project and other Pages

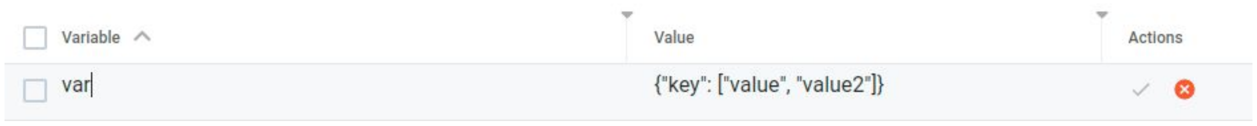
Modal dialog are redesigned and improved over all application.



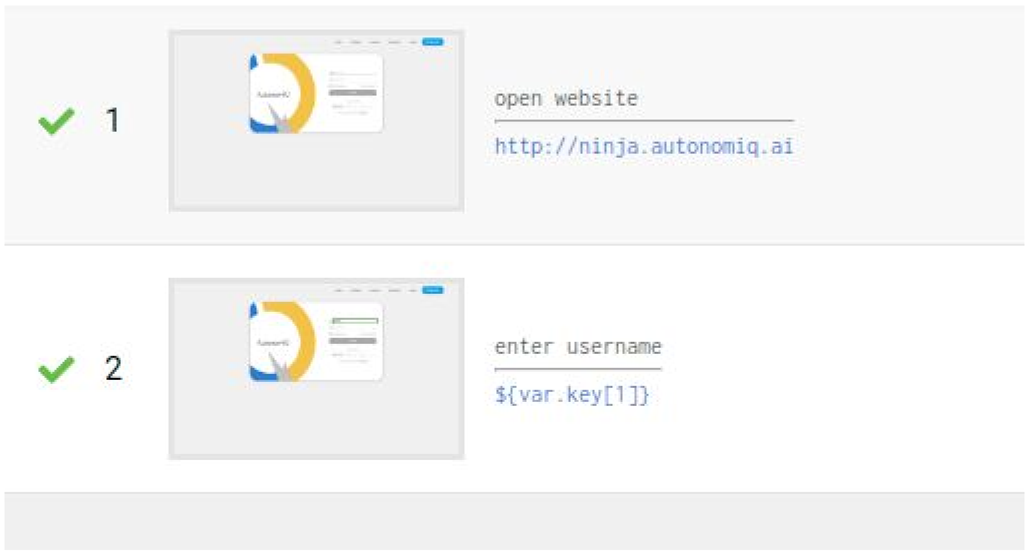
AutonomIQ

Set Variable Value Formatting and access Variable in the list

We can store variable as a list and fetch based on index starting as 1



Variable	Value	Actions
var	{"key": ["value", "value2"]}	✓

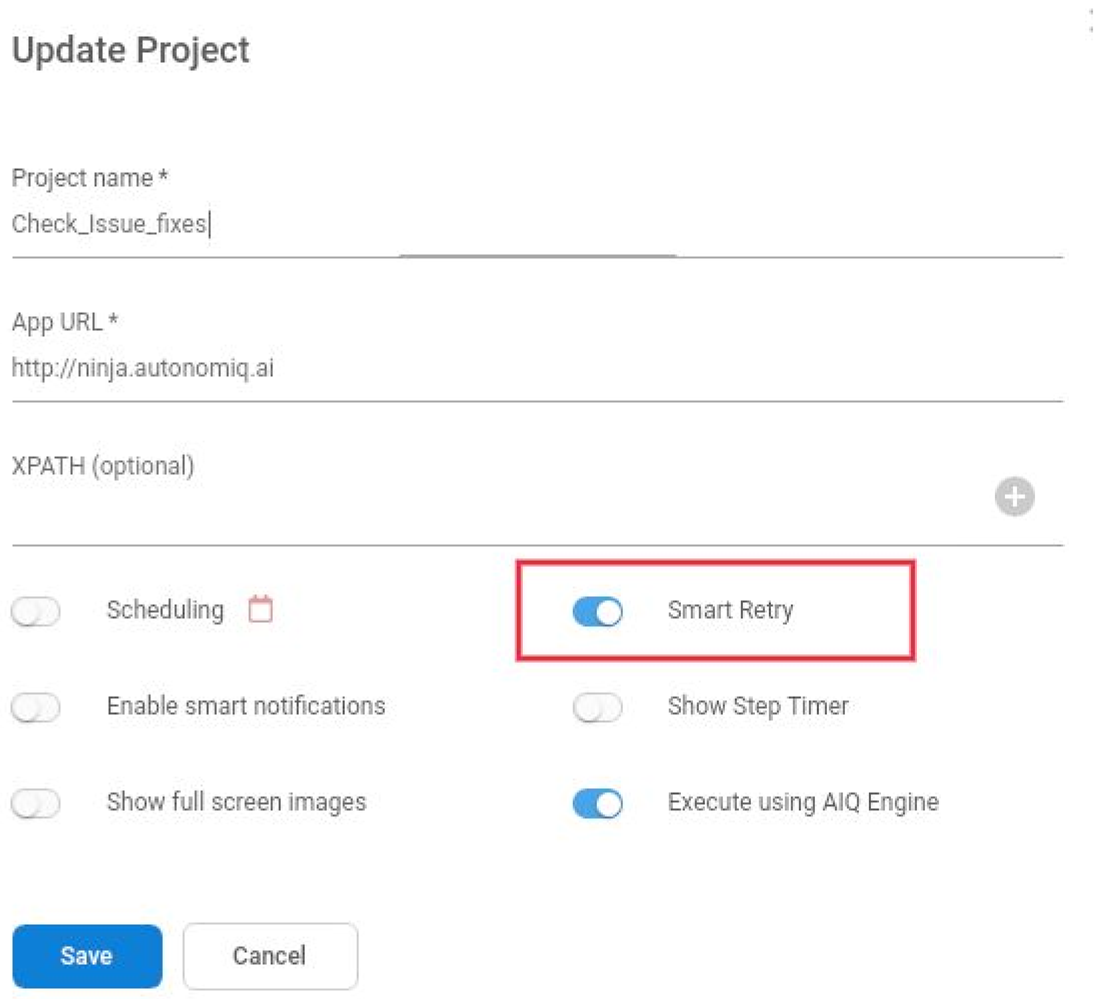


1 open website
http://ninja.autonomiq.ai

2 enter username
${var.key[1]}

AutonomIQ

Smart Retry Timeout



Update Project

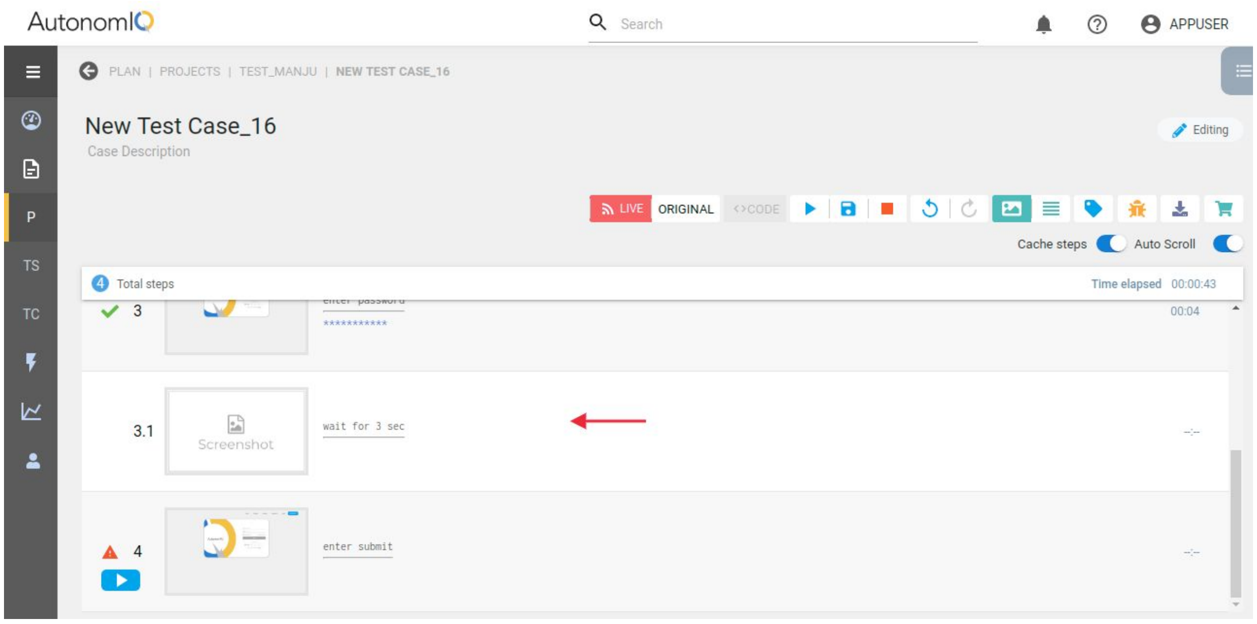
Project name *
Check_Issue_fixes

App URL *
http://ninja.autonomiq.ai

XPATH (optional)

Scheduling: off
Smart Retry: on
Enable smart notifications: off
Show Step Timer: off
Show full screen images: off
Execute using AIQ Engine: on

Enable Smart Retry from Update Project page

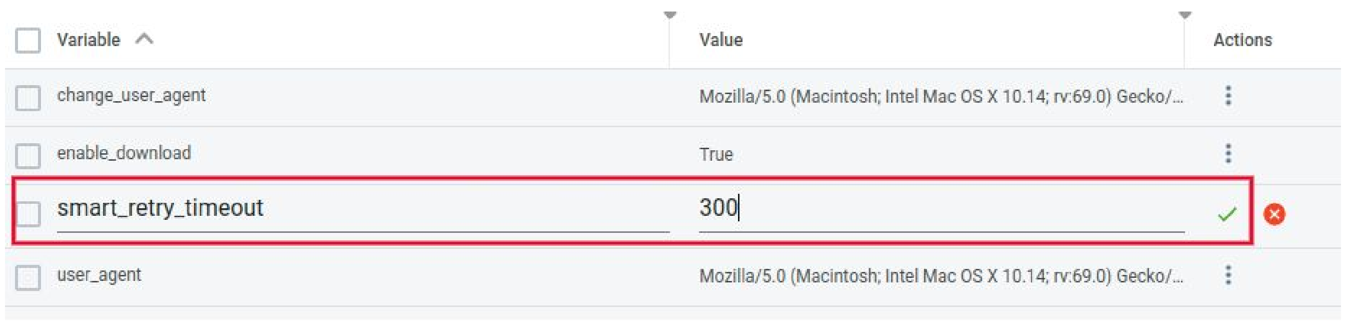


When the project is in the smart retry mode, and suppose test steps fails at step 4, smart retry button will be visible on step 4, and if user edited/added in between

eg: at step 3.1 now click on smart retry icon. the step start generating from 3.1

Smart Retry Timeout (Configurable)

We have a variable called "smart_retry_timeout" we can set the number of minutes for smart retry.. Default value will be 2 minutes, but if user changes variable, it can be whatever the number of minutes the user wants.

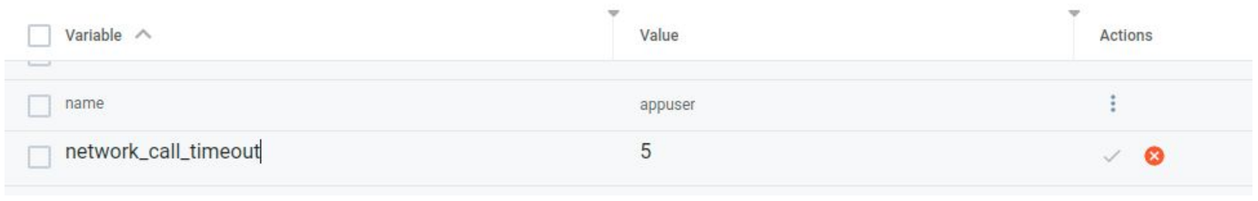


Variable	Value	Actions
change_user_agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:69.0) Gecko/...	⋮
enable_download	True	⋮
smart_retry_timeout	300	✓
user_agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:69.0) Gecko/...	⋮

We can specify a value in seconds, here 300 secs will make smart retry button visible for 5 min so user get enough time to debug the error thrown step

Network Call Timeout

Network call timeout feature will wait for network API calls to get over. This is to ensure that page has loaded properly. By default we have Selenium waits but sometimes it does not give reliable results so using network calls feature we wait for request calls to get over and ensure that page loading has been completed. To enable network call feature, we need to set a variable named ${network_call_timeout} in the variables tab and assign some time (in seconds).

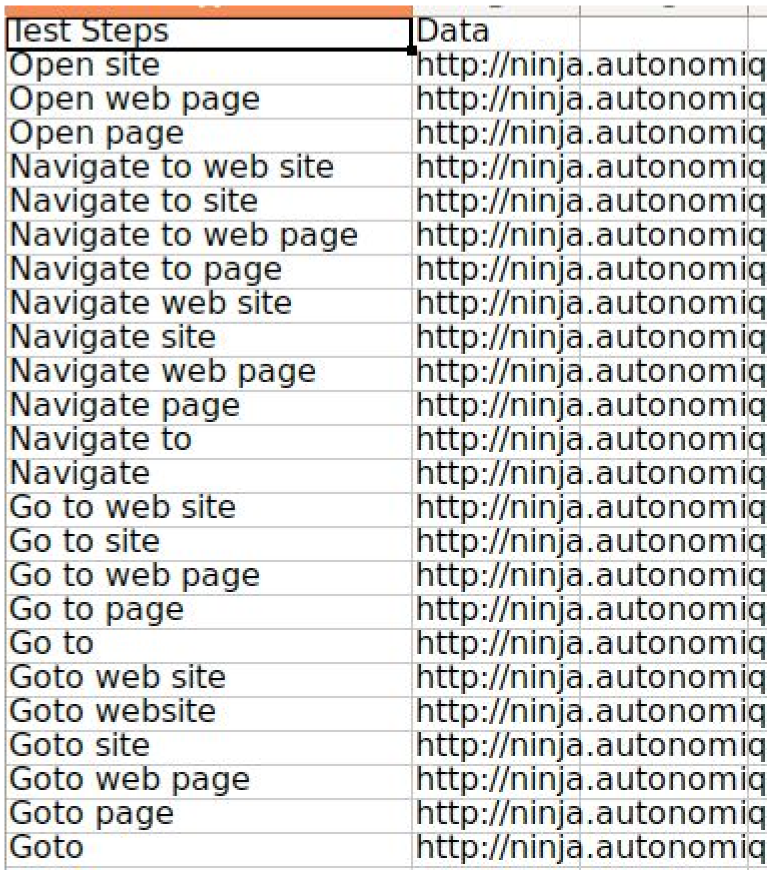


Variable	Value	Actions
name	appuser	⋮
network_call_timeout	5	✓

AutonomIQ

New Instruction Support

Add more synonyms for Open website



Test Steps	Data
Open site	http://ninja.autonomiq
Open web page	http://ninja.autonomiq
Open page	http://ninja.autonomiq
Navigate to web site	http://ninja.autonomiq
Navigate to site	http://ninja.autonomiq
Navigate to web page	http://ninja.autonomiq
Navigate to page	http://ninja.autonomiq
Navigate web site	http://ninja.autonomiq
Navigate site	http://ninja.autonomiq
Navigate web page	http://ninja.autonomiq
Navigate page	http://ninja.autonomiq
Navigate to	http://ninja.autonomiq
Navigate	http://ninja.autonomiq
Go to web site	http://ninja.autonomiq
Go to site	http://ninja.autonomiq
Go to web page	http://ninja.autonomiq
Go to page	http://ninja.autonomiq
Go to	http://ninja.autonomiq
Goto web site	http://ninja.autonomiq
Goto website	http://ninja.autonomiq
Goto site	http://ninja.autonomiq
Goto web page	http://ninja.autonomiq
Goto page	http://ninja.autonomiq
Goto	http://ninja.autonomiq

Switch to Alert box and save the Alert

Since few releases we have instructions

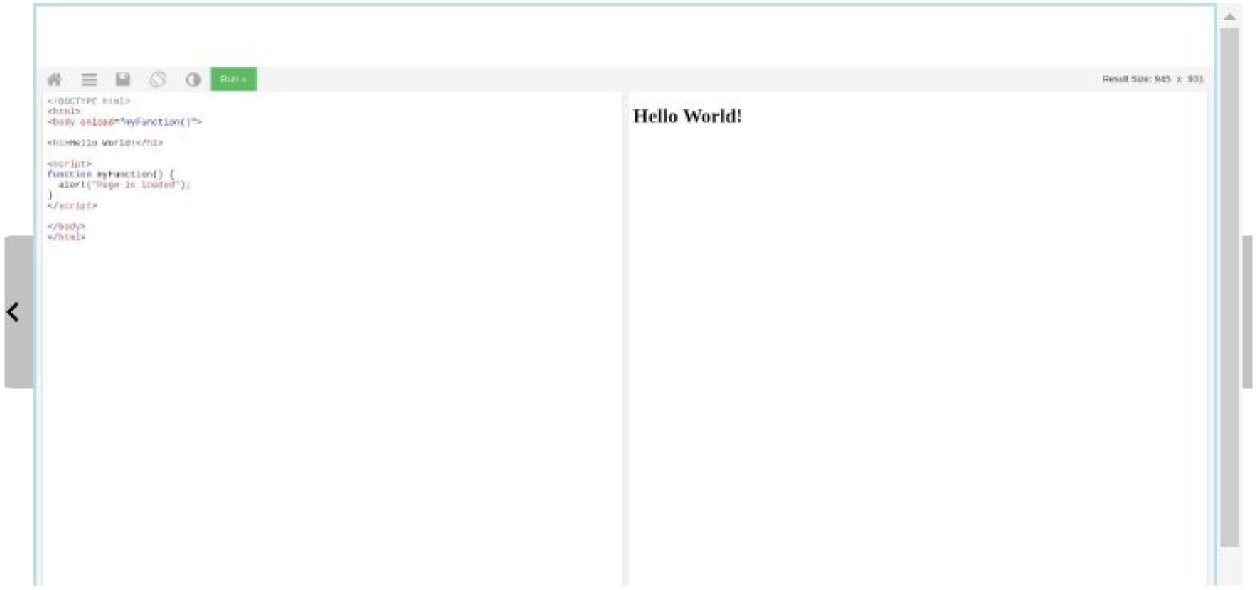
```
switch to alert box and save message as alert_set1
switch to alert and click on ok
```

Now we also support instruction

```
switch to alert box and save message as alert_set2 and click OK
```

✓ 2 switch to alert box and save message as alert_set3 and click OK

Show full screen images

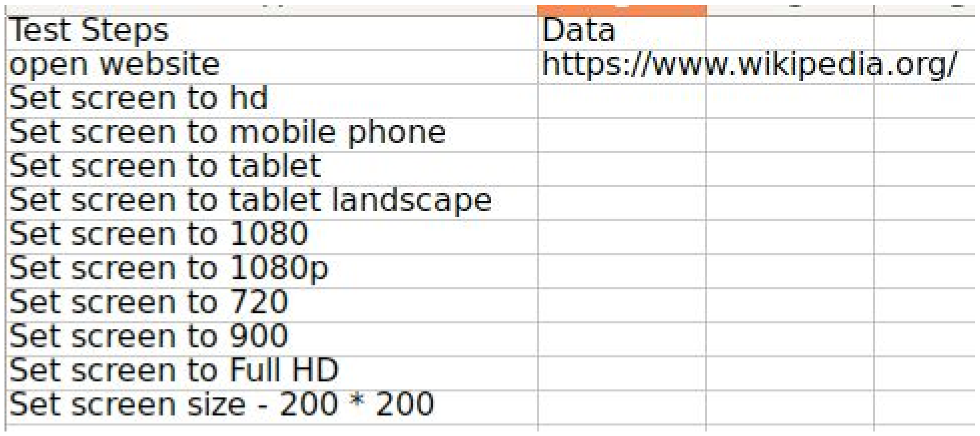


```
<!DOCTYPE html>
<html>
<body onload="myFunction()">

<h1>Hello World!</h1>

<script>
function myFunction() {
  alert("Page is loaded");
}
</script>

</body>
</html>
```

Hello World!

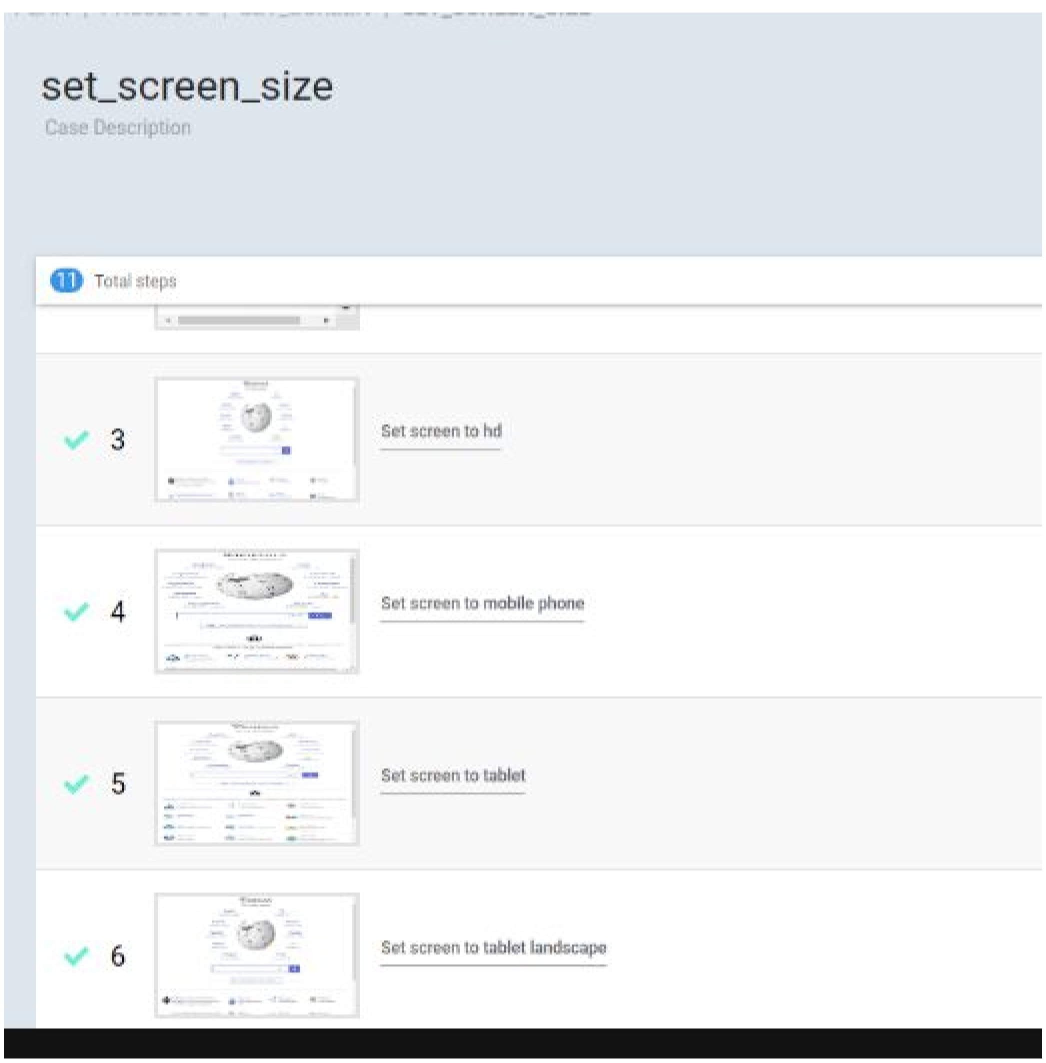
Set screen size to Standard Resolution

User can set the screen to standard resolution with following instructions



Test Steps	Data		
open website	https://www.wikipedia.org/		
Set screen to hd			
Set screen to mobile phone			
Set screen to tablet			
Set screen to tablet landscape			
Set screen to 1080			
Set screen to 1080p			
Set screen to 720			
Set screen to 900			
Set screen to Full HD			
Set screen size - 200 * 200			

AutonomIQ


set_screen_size

Case Description

11 Total steps

✓ 3 Set screen to hd

✓ 4 Set screen to mobile phone

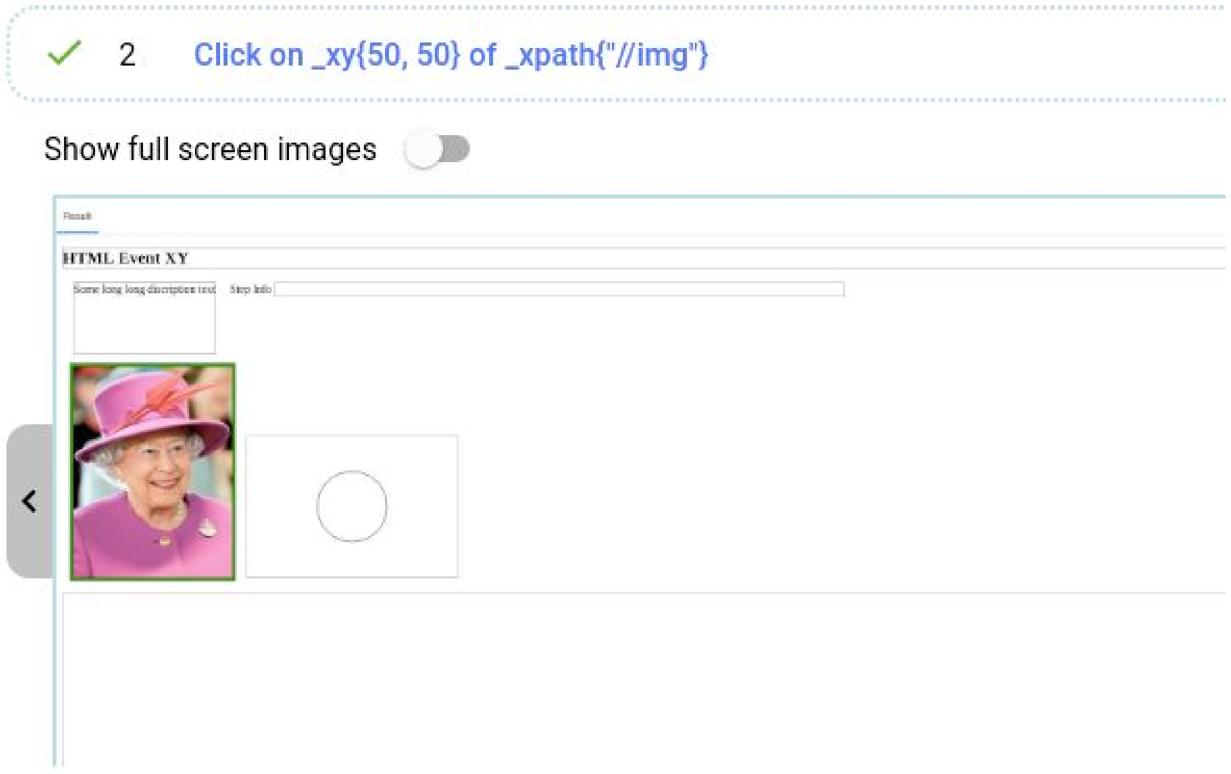
✓ 5 Set screen to tablet

✓ 6 Set screen to tablet landscape

AutonomIQ

Command to click at (x,y)

We can now give XY coordinate on any HTML node and ask system to click at that position. It uses the syntax _xy{ }


2 Click on _xy{50, 50} of _xpath{"//img"}

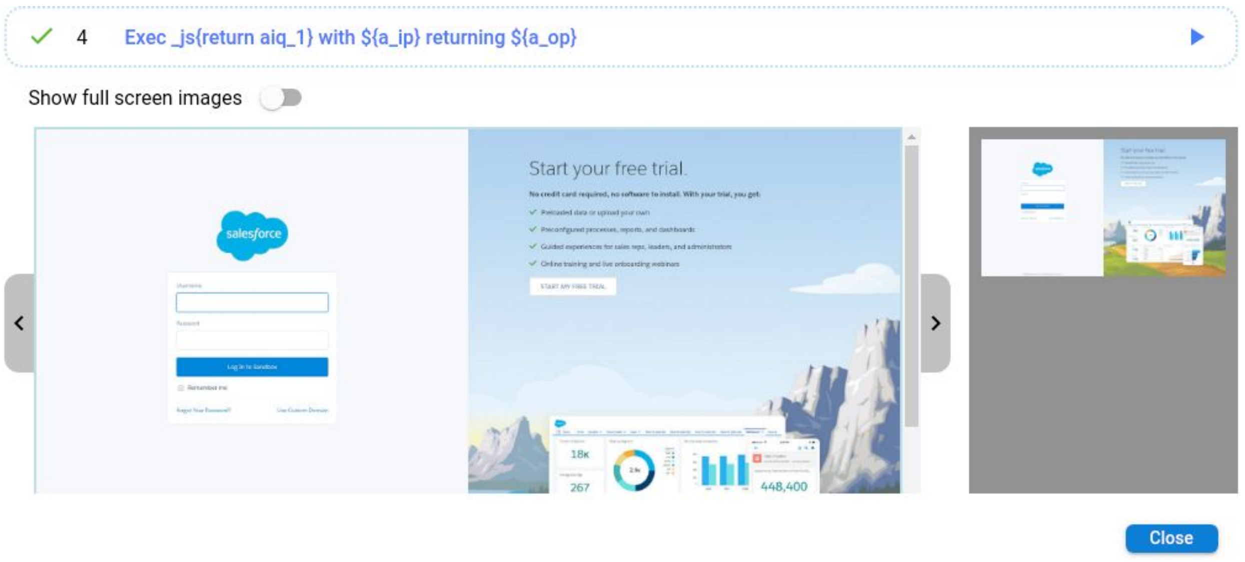
Below are the sample instructions that are supported

```
Click on _xy{20, 30} of _css{#some_html_node_id}
Hover on _xy{20, 30} of _xpath{//img[@id='some_html_node_id']}
Double Click on _xy{20, 30} of "Photo of Eiffel Tower”
```

AutonomIQ

Bug Fixes

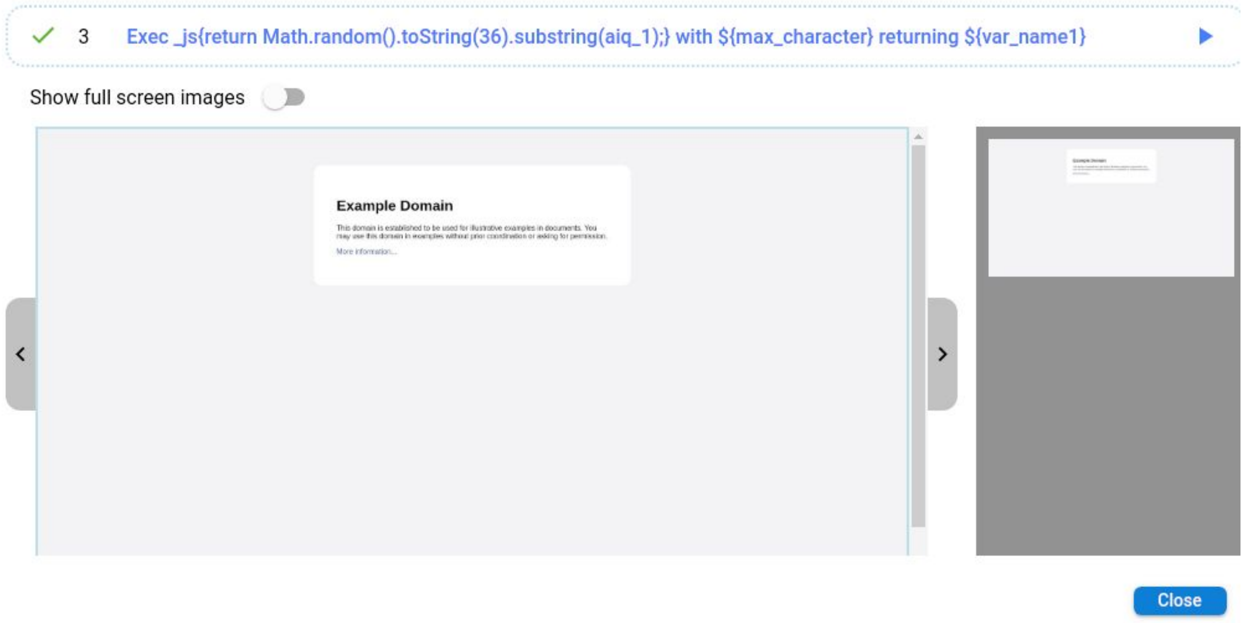
Basic _py, _js instruction works now



✓ 4 Exec _js{return aiq_1} with ${a_ip} returning ${a_op}

Show full screen images

salesforce
Log In to Sandbox
Remember me
Start your free trial.
No credit card required, no software to install. With your trial, you get:
Preloaded data or upload your own
Preconfigured processes, reports, and dashboards
Guided experiences for sales reps, leaders, and administrators
Online training and live onboarding webinars
START MY FREE TRIAL
18K
267
448,400
Close



✓ 3 Exec _js{return Math.random().toString(36).substring(aiq_1);} with ${max_character} returning ${var_name1}

Show full screen images

Example Domain

This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.

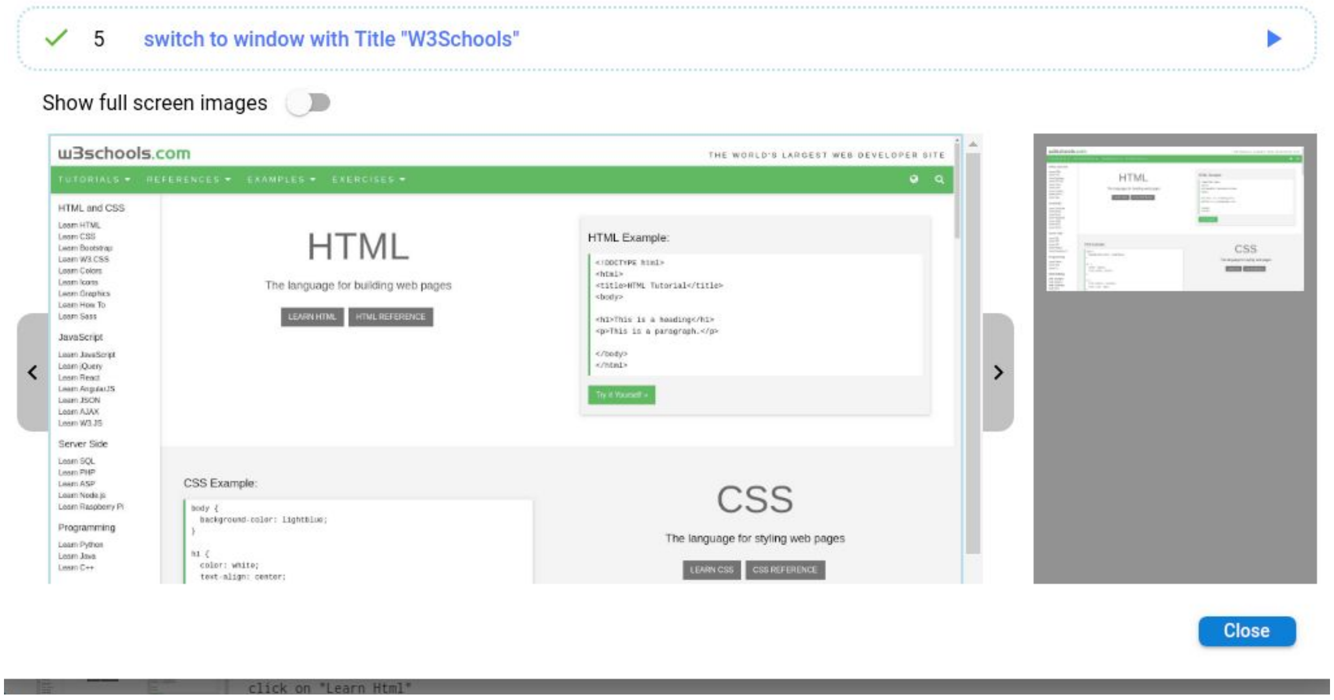
More information...

AutonomIQ

Switch with title instruction

We can now switch to a window by providing its title.

switch to window with title “title1”



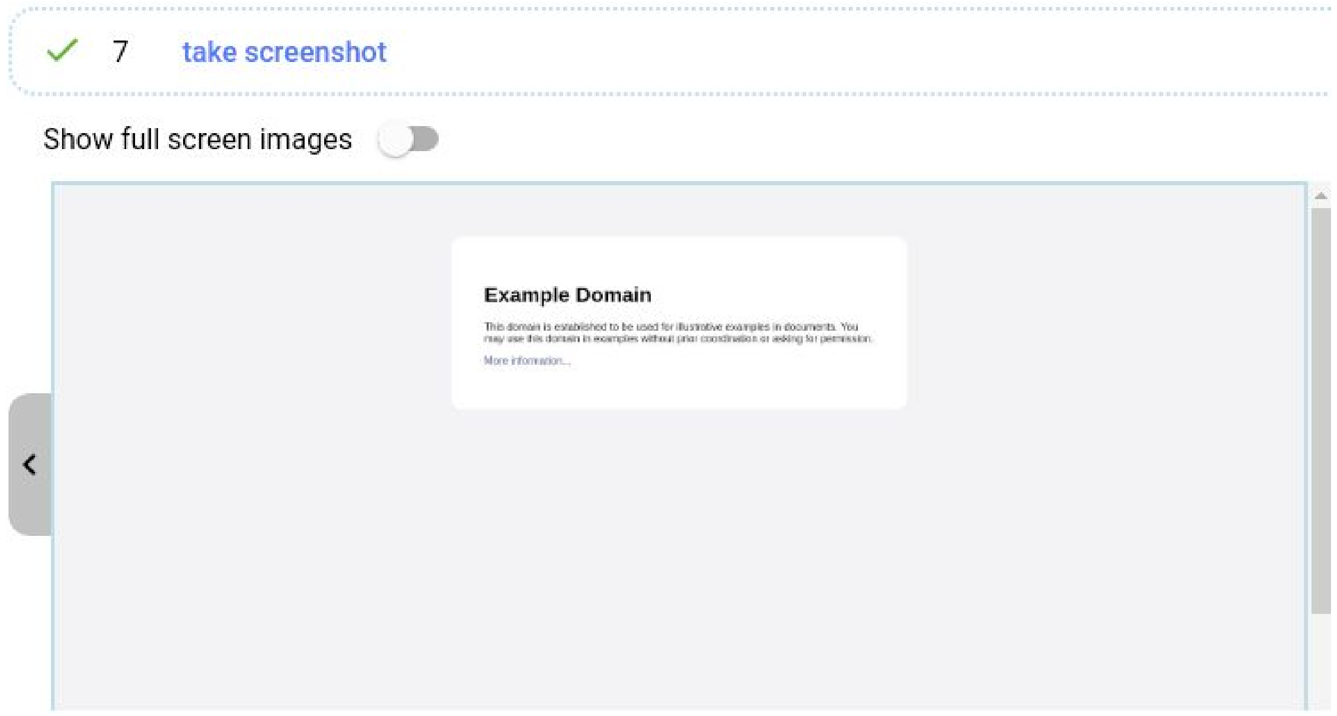
5 switch to window with Title "W3Schools"

Show full screen images

click on "Learn Html"

AutonomIQ

Take Screenshot


✓ 7 take screenshot

Show full screen images

Example Domain

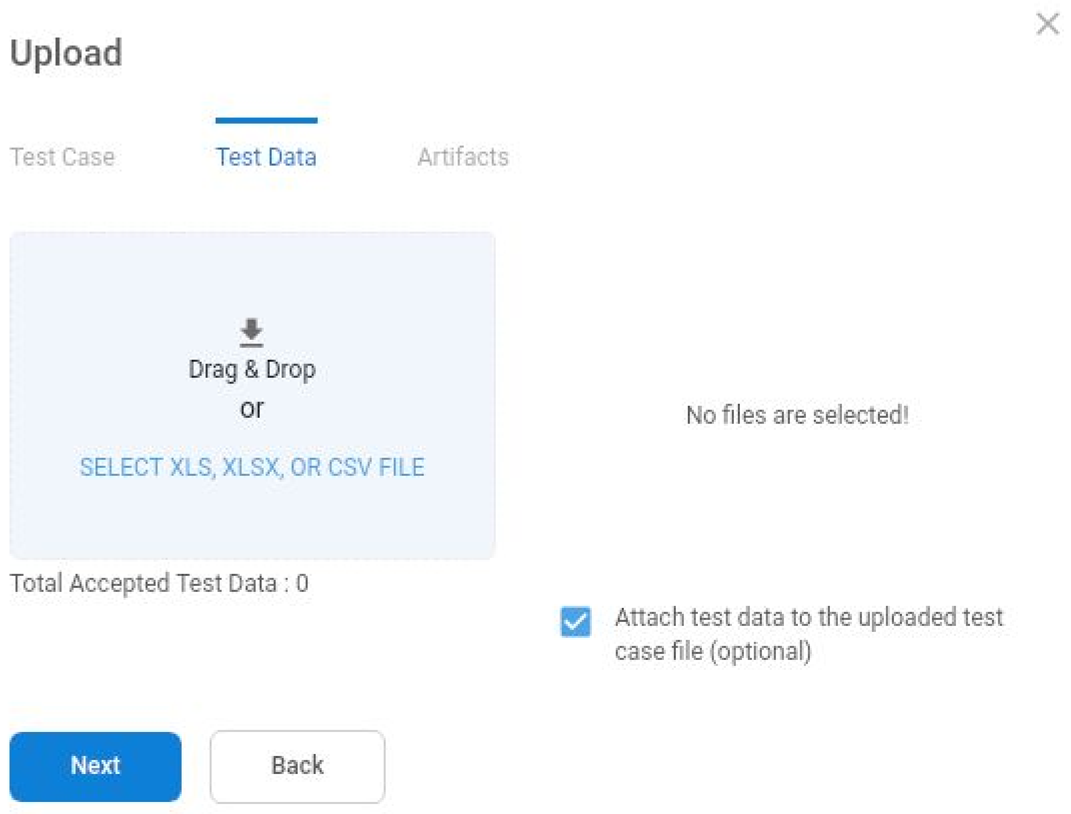
This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.

More information...

Statements “take screenshot” and “capture screenshots” works now

AutonomIQ

Same model size in upload



Upload

Test Case | Test Data | Artifacts

Drag & Drop
or
SELECT XLS, XLSX, OR CSV FILE

No files are selected!

Total Accepted Test Data : 0

✓ Attach test data to the uploaded test case file (optional)

Next | Back

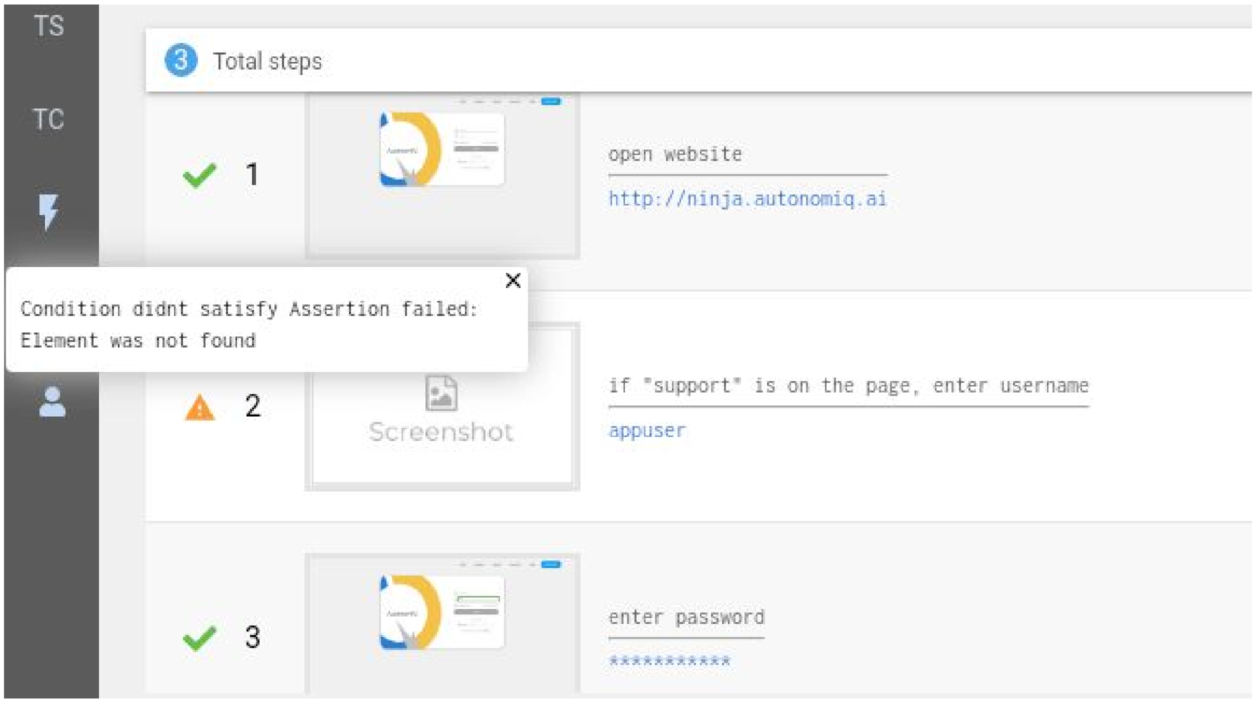
Before this model size was not even when clicking Next

AutonomIQ

Dont discover when if condition don't satisfy

When the if condition won't satisfy, it won't discover the next instruction .

ex. For instruction if “login” is on the page, enter username if login is not there , we won't discover enterable username at all. And system will simply just move to next instruction



1 open website
http://ninja.autonomiq.ai

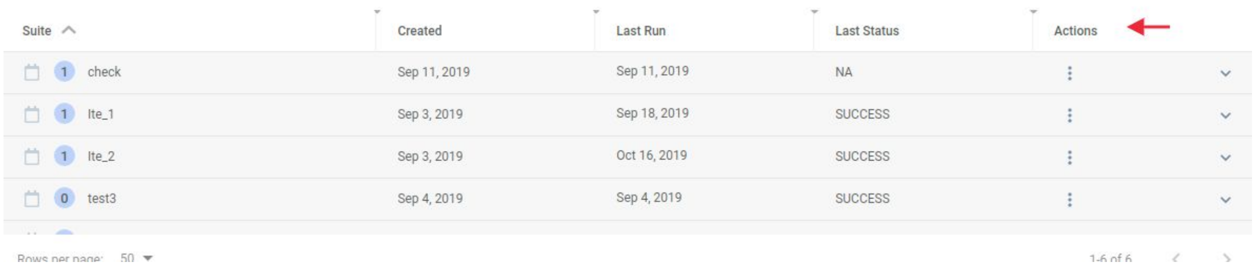
Condition didnt satisfy Assertion failed: Element was not found

2 if "support" is on the page, enter username
appuser

3 enter password
****

Table header out of scroll

Now header will stay when the list is scrolled



Suite	Created	Last Run	Last Status	Actions
1 check	Sep 11, 2019	Sep 11, 2019	NA	⋮
1 Ite_1	Sep 3, 2019	Sep 18, 2019	SUCCESS	⋮
1 Ite_2	Sep 3, 2019	Oct 16, 2019	SUCCESS	⋮
0 test3	Sep 4, 2019	Sep 4, 2019	SUCCESS	⋮

Upload file with Artifact extension

Earlier below instruction i.e artifact for upload file name is allowed without giving file name extension

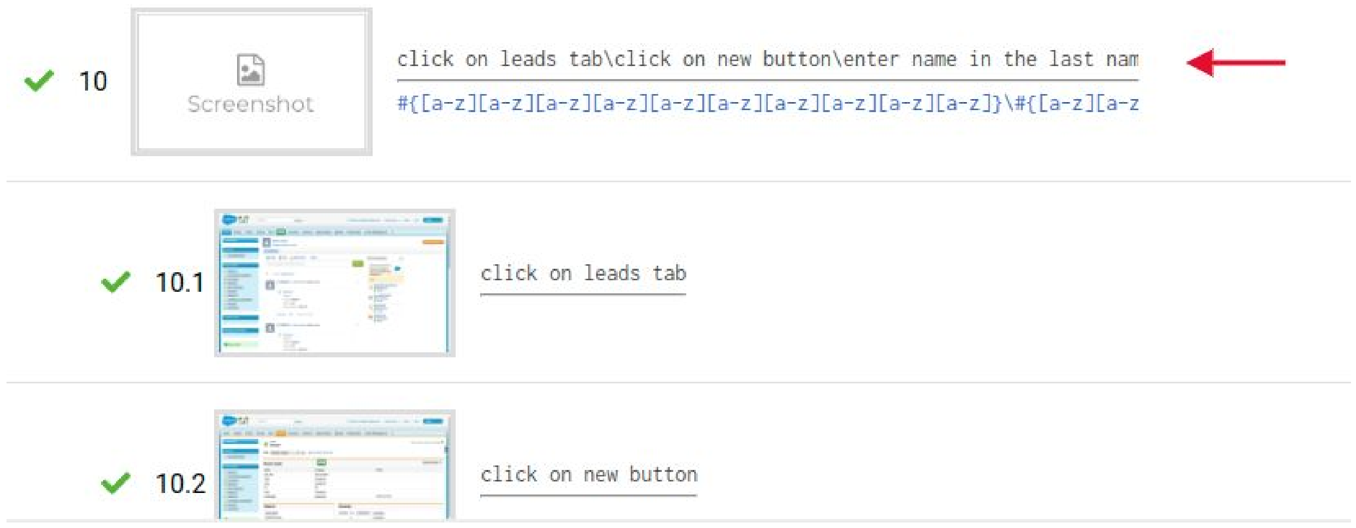
upload file to "upfile"	ArtifactForUploadFile

Now the same is supported only by giving file name extension refer below

upload file to "upfile"	ArtifactForUploadFile.xls

If we allow filenames without extensions, and if multiple files of the same name but different extensions are uploaded, there’s no way for Autonomiq’s script generation engine to identify the right file to be used.

The Compound statement works after Add/Edit the test step



✓ 10 Screenshot

click on leads tab\click on new button\enter name in the last nam

#{[a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z][a-z]}\#{[a-z][a-z

✓ 10.1 click on leads tab

✓ 10.2 click on new button

The Compound statement creation during Test Step Add/Edit when separated with "." NLP break's down the test steps accordingly

Performance Improvement

We have reduced the message size to improve performance and fix message passing for huge script, Check whether content first page is updated

Known Bugs

1. When bulk uploading test cases, the script generation for uploaded tc's is not supposed to start automatically. However one of the test cases will display the status as In Progress, although the script is not generating.
2. The alert box/pop-up won't be captured in the screenshot if it is present in the page/application at the current step.

Enhancement

1. Provide support for instruction "set screen size 600* 600". Now the same works when we give "set screen size - 600* 600"

AutonomIQ

Version Details

Following are the version changes in the version 5.1

Mozilla Firefox 62.0.3
Geckodriver 0.25.0
Google Chrome 75.0.3770.80
ChromeDriver 75.0.3770.90
Selenium 3.12.0

Following are the version changes in the version 5.0

Mozilla Firefox 62.0.3
Geckodriver 0.20.1
Google Chrome 75.0.3770.80
ChromeDriver 75.0.3770.90
Selenium 3.8.0

AutonomIQ

Optional Arguments

Optional arguments can be provided to a test step as described below.

Ignore Alert

By default, Autonomiq will check if a browser alert is present on the screen before interacting with any element on the screen. If an unhandled alert is present (alerts can be handled by – switch to alert and click on OK/Cancel), it'll purposely fail the test step with an error message stating that the alert is unhandled. If the user doesn't want for the test step to fail, they can use the ignoreAlert option as shown below

Click on "login" button –ignoreAlert

Dynamic Xpath

By default, Autonomiq caches xpaths for every test step so that subsequent script generations will be faster. However, if the user doesn't want to use the cached xpath for a certain step, they can provide the dynamicXpath option as shown below

Click on ${order_id} --dynamicXpath

Note: If a certain xpath is not valid due to it being dynamic or an application change, it will be auto-healed which guarantees that the plain English step will not fail due to invalid xpaths.

AutonomIQ

Dealing with disabled elements (visually grayed out)

By default, Autonomiq will only interact with elements that are enabled. If the user wants to interact with a disabled element, they can use the Force option as shown below

Force click on "login"
where login button is grayed out.

Using Actions chain click

By default, Autonomiq uses selenium click and if selenium click fails, it'll switch to javascript click. However, if the user wants to specifically use action-chain click, they can provide it as shown below

Click on "login" --moveAndClick

Provide spinner/progress bar information

If the application under test has progress bars/spinners as a part of the UI design, Autonomiq provides the capability for users to specify the spinner information as a variable as shown below under "variables" tab. Once this information is provided, Autonomiq will dynamically wait until the progress bar/spinner disappears before proceeding with the next step.

Variable name : spinner_xpath
Variable value : xpath_of_the_spinner